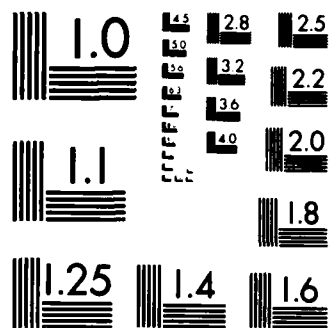END

FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A158 476

FERMI:
A Flexible Expert Reasoner
with
Multi-domain Inferencing

DEPARTMENT
of
PSYCHOLOGY

DTIC
ELECTE
AUG 2 9 1985
S   D
A

DTIC FILE COPY

Carnegie-Mellon University

85 8 26 064

FERMI:
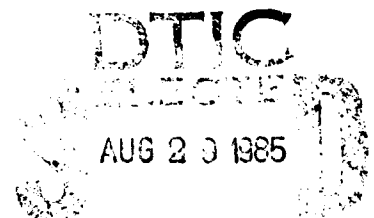# A Flexible Expert Reasoner
with
Multi-domain Inferencing

Jill H. Larkin
Department of Psychology. Carnegie-Mellon University

Frederick Reif
Department of Physics and School of Education
University of California, Berkeley

Jaime Carbonell and Angela Gugliotta
Department of Computer Science. Carnegie-Mellon University

29 July 1985

Running Head: Flexible Expert Reasoner

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
Unclassified	None

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT
	Distribution according to ONR distribution
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	list; availability unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
Carnegie-Mellon University		

6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
Psychology Department 5000 Forbes Avenue Pittsburgh, PA 15213	

8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
Office of Naval Research		

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
Personnel Training & Research Programs Office of Naval Research (Code 442 PT) Arlington, VA 22217	PROGRAM ELEMENT NO.	PROJECT NO. N-00014-82-C-0767	TASK NO. 667-494	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

FERMI: A Flexible Expert Reasoner with Multi-domain Inferencing

12. PERSONAL AUTHOR(S)
Larkin, Jill H., Reif, Frederick, Carbonell, Jaime, and Gugliotta, Angela

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 1982 TO 1986	1985 July 13	7-8

16. SUPPLEMENTARY NOTATION

This work was supported by the Office of Naval Research Con. # N-00014-82-0767

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT

Expert reasoning combines voluminous domain-specific knowledge with more general factual and strategic knowledge. Whereas expert system builders have recognized the need for specificity and problem-solving researchers the need for generality, few attempts have been made to develop expert reasoning engines combining different kinds of knowledge at different levels of generality. This paper reports on the FERMI project, a computer-implemented expert reasoner in the natural sciences that encodes factual and strategic knowledge in separate semantic hierarchies. The principled decomposition of knowledge according to type and level of specificity yields both power and cross-domain generality, as demonstrated in FERMI's ability to apply the same principles of invariance and decomposition to solve problems in fluid statics, DC-circuits, and centroid location. Hierarchical knowledge representation and problem-solving principles are discussed, and illustrative problem-solving traces are presented.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☑ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

FERMI: A Flexible Expert Reasoner with Multi-domain Inferencing

Jill H. Larkin Department of Psychology, Carnegie-Mellon University

Frederick Reif Department of Physics and School of Education University of California, Berkeley

Jaime Carbonell and Angela Gugliotta Department of Computer Science, Carnegie-Mellon University

1. Introduction

Most computer-implemented expert systems, and many humans, encode their knowledge so that information of varying generality is indiscriminately mingled. As a result, potentially general knowledge encoded in one domain cannot readily be used in others -- making it difficult to extend knowledge or to respond flexibly to novel situations. Furthermore, this undifferentiated mingling of general and specific knowledge makes it difficult for a system to explain its reasoning since it is unable to relate features of various problems to a limited amount of general knowledge.

This paper describes a prototype computer-implemented problem-solving system that aims to overcome the preceding difficulties by encoding its declarative and procedural information hierarchically at the appropriate level of generality. In particular, the system includes two related hierarchies, one of scientific principles and one of problem-solving methods. We call the system FERMI, an acronym for Flexible Expert Reasoner with Multiple-domain Inferencing, and also a tribute to the physicist Enrico Fermi who was well-known for his powerful abilities to reason from general principles in multiple domains in the natural sciences.

FERMI initially solved problems in one domain (a subset of fluid statics). Its general knowledge has since then proved readily extensible, with little addition of problem solving code, to two quite unrelated domains (to DC circuits and to computations of centers of mass). It is our hope that FERMI will ultimately demonstrate how expert systems can be built to achieve much greater power by judiciously separating knowledge according to levels of generality. Moreover, the separation of strategic knowledge (e.g., problem-solving methods) and factual knowledge (e.g., scientific relations such as invariance, decomposition, and equilibrium), provide far more generative power than would present in a system encoding explicitly

combinations of various types of knowledge.

In addition to its problem solving power, we expect FERMI to provide hypotheses of how skilled human experts separate knowledge according to its generality. Finally, FERMI should allow us to test how general knowledge can be communicated to humans so as to help them flexibly learn and use large amounts of knowledge.

The next few paragraphs give a brief overview of the deficiencies exhibited by many current expert systems, of the considerations underlying the design of FERMI, and of the general knowledge and methods presently incorporated in FERMI. The subsequent sections discuss in greater detail the structure of FERMI, the diverse problems which our prototype version of FERMI has been able to solve on the basis of its limited general knowledge, and future extensions of FERMI which promise to lead to much greater problem-solving power and also to some educational applications.

1.1. Common Deficiencies of Expert Systems

Rule-based expert systems often mingle indiscriminately all kinds of strategic or factual knowledge, irrespective of cross-domain generality. General principles and domain-specific instances, widely applicable methods and particular inferences, are all woven into an amorphous set of rules comprising the "knowledge base" of the system. It is probably not surprising that this situation arises both in human learners and in the development of expert systems. Indeed, knowledge is usually acquired in a specific context by observing and encoding the efforts or explanations of some "informant" (e.g., an expert consultant, a textbook, or a teacher). In this specific context, this informant applies both general and specific knowledge. The learner or the knowledge engineer building the expert system (neither of whom are domain experts) have no way to assess the potential generality of the knowledge being exhibited. Therefore, he or she naturally encodes a mixture of knowledge applicable to the specific context. In fact, this approach can work quite well in domains sufficiently focused that broad use of general knowledge is not required (Shortliffe, 1976, Feigenbaum, 1971, McDermott, 1980, Mcdermott, 1982).

Mixing knowledge of different degrees of generality is, however, not only unaesthetic and unprincipled; it also makes it almost impossible to apply general knowledge to other domains, no matter how similar to the present expertise. There is no way to identify, much less extract and reason from, the general knowledge. Correspondingly, the following problems arise almost universally both for human learners (Reed, Ernst & Banerji, 1974, Brown & Campione, 1984) and for computer-

implemented expert systems.

- *Brittleness.* The system or learner cannot readily respond to minor variations leading to situations even slightly different from those for which the knowledge was originally encoded.

- *Lack of generalizability or transfer.* The system or learner cannot readily extend his or her knowledge beyond the bounds of that originally available, nor extend it to other domains. Dealing with somewhat different circumstances requires both the addition of new domain-specific knowledge as well as the recoding of more general principles.

- *Lack of explanatory power.* The system cannot explain its reasoning in a manner that human beings can easily understand. A major reason is that general knowledge, shared by the system and the human, cannot provide an explicit basis for explanation, but becomes submerged in a morass of specific details.

1.2. Design of the FERMI System

FERMI is a working prototype of an expert reasoner that aims to overcome the preceding deficiencies by explicitly separating knowledge according to its level of generality. The system employs a standard artificial-intelligence method for building knowledge-representation systems. In particular, FERMI uses schemas, (Brachman, 1979, Minsky, 1975, 1975, Fox, 1979, Bobrow, 1977)data structures composed of *slots* and corresponding *fillers*. Each schema provides a structure for storing related knowledge. Any slot in a schema may have associated information about how the slot may be filled (e.g., default values and constraints). Slots in FERMI may also have associated *pullers*, i.e., pieces of code to be implemented whenever the system needs to fill the slot about which it has no stored information. [For example, a puller associated with the *mass slot* of an object schema might check whether the average density and volume of the object are known. If so, the puller would multiply these quantities and place the result as the filler of the mass slot.] Pullers have in different guises have occured throughout the AI literature, perhaps first emerging as *if-added demons* in the CONNIVER system (Sussman, 1975), and continuing through KRL (Bobrow, 1977) and SRL (Wright & Fox, 1983, 1983).

FERMI's schemas are connected in hierarchies by *isa* links indicating class membership. When a schema A is connected by an *isa* link to a second schema B, then A inherits automatically all the contents (slots, fillers, associated information and pullers) from the schema B. The *isa* relation is transitive; in other words, if A *isa* B and B *isa* C, then B inherits directly the contents of C, and A inherits from B both

the original contents of B and all the knowledge that B inherited from C. Inheritance occurs along any chain of *isa* links. Because of this inheritance, knowledge common to a variety of schemas need be encoded only once. The inheritance structure is thus similar to that of tangled semantic networks (Bobrow, 1977, Brachman, 1979, Fahlman, 1979), with the added proviso that constraints, defaults, and pullers are also inherited.

FERMI is also based on research on how information is structured in the physical sciences (Chi, Feltovich, & Glaser, 1981, Reif & Heller, 1982). Physical scientists can identify general principles and problem-solving methods (e.g., energy principles or decomposition methods) as well as specific instantiations (e.g., decomposition of vectors into components). They can also distinguish between more and less general principles or methods. (For example, the statement "path integrals of scalar-field differences are path independent" is quite general, while the statement "pressure drop in a static fluid is path independent" is specific to the domain of fluid statics.) Reflecting experts' knowledge of more and less general principles and methods, FERMI has two distinct schema hierarchies, one encoding scientific principles of different levels of generality, and the other encoding problem-solving methods of different levels of generality. These two hierarchies interact to produce FERMI's abilities.

A system like FERMI, designed according to the preceding guidelines, should have the following advantages: (1) It should overcome many of the deficiencies of more traditional expert systems by being more robust, more readily extensible to other domains, and more readily able to explain its reasoning to humans. (2) Such a system should also provide a concrete testable model simulating the knowledge structures of sophisticated human experts capable of flexibly using and generalizing large amounts of knowledge. In this respect, FERMI might serve as a particularly interesting computer-implemented model of the kind used in recent years to understand the psychology of expert performance (Larkin, McDermott, Simon, & Simon, 1980a, Larkin, McDermott, Simon, & Simon, 1980b, Simon & Simon, 1978). (3) A FERMI-like system could serve as the knowledge base for a powerful instructional system that, unlike many human instructors, would guide human learners in acquiring knowledge usefully structured according to levels of generality.

2. Principles in FERMI

One can envision a system like FERMI encoding a variety of general relations valid for reasoning about the natural sciences, such as: *decomposition, invariance, equilibrium, etc.* The currently completed prototype of FERMI requires only the first

two of these principles, and their associated methods, to solve a variety of problems.

2.1. Decompoition

The first of these principles states that certain quantities are decomposable in the following sense: Consider a quantity Q which is a function of some entity E, a relation which can be formally written as $Q = Q(E)$.   (For example, the quantity Q might be a pressure drop which is a function of the path joining two points in a liquid.) Suppose also that this entity E can be conceived as consisting of subentities E_i.   (For example, a path can be conceived to consist of subpaths or segments which collectively make up the original path.)   A corresponding quantity $Q(E_i)$ is then associated with each such subentity E_i.   (For example, there is a pressure drop associated with each component segment of a path.)   If the quantity Q is decomposable with respect to the entity E, then the value of the total quantity Q associated with E is simply equal to some specified combination function applied to the values $Q(E_i)$ of Q associated with the individual components E_i of E. (For example, the pressure drop associated with a path is simply equal to the arithmetic sum of the pressure drops associated with all the individual segments of this path.)

The decomposition principle can be summarized in the following equation:

$$Q(E) = \Sigma_i \, Q(E_i),$$

where and $\{E_i\}$ are a set of entities that can be composed into the original entity E. More generally the summation might be replaced by other composition functions (e.g., multiplication, weighted addition).

The general 'decomposition method' associated with this principle applies to all decomposable quantities.   It specifies the following general procedure to find the value of a quantity from the values associated with its components: (1) If a quantity Q is decomposable with respect to an entity E, find a decomposition into component entities E_i such that each associated value $Q(E_i)$ is less difficult to compute than Q. (2) Compute the values $Q(E_i)$ associated with all these component entities. (3) Combine these values by using the specified combination function.

The decomposition principle and associated decomposition methods apply to functions of many types of entities. For example, decomposition applies to pressure drops or potential drops as functions of paths, to areas or centers of mass as functions of regions, and to temporal functions expressed as functions of frequency. As discussed later, the decomposition meth' ds can also use different strategies

(iterative decomposition or recursive decomposition) for choosing useful component entities.

2.2. Comparison of Invariants

The second general principle used by FERMI is the invariance principle which specifies that a quantity Q is invariant in the following sense: Suppose that Q is a function of an entity E or set of such entities. (For example, the quantity Q might be the energy of a particle and would then be a function of the position and velocity of this particle.) Then Q is invariant with respect to E if Q remains the same if E is changed. (For example, under appropriate conditions, the energy of a particle remains invariant when its position and velocity change.) The equation

$$Q(E_i) = \text{constant, for all } E_i$$

expresses the invarience principle, where E_1 and E_2 are two values of the entity E.

The following method ("comparison of invariants") is used in conjunction with the invariance principle: (1) If a quantity Q invariant with respect to an entity E, select two values (E_1 and E_2) of E relevant to quantities mentioned in the problem. (2) Compute and equate the expressions Q_1 and Q_2 for the values of Q associated with E_1 and E_2. The result of this method is an equation relating quantities of interest in the problem. [For example, the energy of a particle can be expressed in terms of its position and velocity. If the particle's energy is invariant, a consideration of the particle at two different times (corresponding to different positions and velocities of the particle) yields an equation relating these positions and velocities.]

2.3. Current implementations of these principles.

The general principles of decomposability and invariance are currently used by FERMI in three domains. The first is the computation of pressure drops in fluids at rest. The second is the calculation of potential drops in direct-current electric circuits. The third is the calculation of centers of mass for planar objects. (The calculation of centers of mass involves composition by a weighted-average operator, unlike the calculation of pressure or potential drops which involves composition by simple scalar addition.) Thus FERMI currently implements two principles (decomposability and invariance) with associated methods, and can use this knowledge in three domains to compute pressure drops, potential drops, and centers of mass.

The next two sections of this paper describe the current implementation of FERMI, first discussing its architecture and then its problem-solving performance in

various domains. The final section discusses the potential power of this approach by describing planned future extensions of FERMI and the far larger range of problems that it should then be able to address with a relatively small number of additional general strategies and principles.

3. Architecture

The first part of this section provides an overview of FERMI's problem-solving capabilities and of the architecture of its knowledge. The remaining parts discuss more technical details of FERMI's implementation.

3.1. FERMI's Knowledge and Capabilities

FERMI currently can find the pressure drop between any two points in a liquid, the potential drop between any two points in an electric circuit, and the position of the center of mass of any planar object decomposable into rectangular parts. To solve such problems in all three of these domains, in both simple and more complex situations, FERMI requires the same general principle (the decomposability principle) and associated methods -- together with some domain-specific knowledge about fluid statics, about electric circuits, and about the definition of center of mass.

FERMI can also solve moderately complex electric-circuit problems by using the preceding general decomposability and invariance principles together with its domain-specific knowledge about electric circuits. In addition, FERMI can use algebra flexibly to solve an equation for any quantity appearing in it and to propagate symbolic algebraic expressions without requiring numerical solutions.

FERMI's knowledge is entirely stored in schemas of various levels of generality. Its general knowledge is stored in general "quantity schemas" and in associated general "method schemas". Its domain-specific knowledge is stored in domain-specific quantity schemas and in associated local methods called "pullers". The following paragraphs describe briefly FERMI's domain-specific and general knowledge, and how they interact to solve various kinds of problems.

3.1.1. Domain-specific knowledge.

For each domain, FERMI has specific knowledge encoded in quantity schemas. We use single quotes to indicate the name of any schema. (For example, 'pressure drop' is the central domain-specific quantity schema used for problems in fluid statics.) We indicate the name of a slot in a schema by using single quotes to enclose the schema name separated by a slash from the name of the particular slot. (For example, 'pressure drop/result' is the "result slot" of the 'pressure drop' schema, i.e.,

the slot which stores the value of the quantity "pressure drop". We omit the schema name, retaining only the slash, if the schema name is obvious from the context (e.g., the previous slot may be simply indicated by '/result'.)

Certain slots in quantity schemas have associated with them domain-specific actions called *"pullers"*. A puller contains procedural knowledge of how to fill the slot when it is empty and no inheritable value is available.

For example, the 'pressure drop/result' slot has a puller that can find the value of the pressure drop when specific conditions (specified in the slot '/computability requirements') are satisfied. This puller encodes the following particular method based on domain-specific knowledge about fluid statics:

If two points are
located one below the other,
separated by a height h,
in a single fluid of density ρ and
not separated by a wall,

then find the pressure drop from the lower to the higher by computing the product
ρgh.

The phrases in italics are the contents of the 'pressure drop/computability requirements' slot. FERMI first checks that these requirements are satisfied. and, if so, proceeds to compute the indicated product.

FERMI uses this puller to solve simple problems like the following:

A beaker is filled with oil (of density 0.8 gm/cm^3). What is the pressure drop from a point at the bottom of the beaker to a point located in the oil 3 cm above the bottom point?

In this example, the puller associated with 'pressure drop/result' first checks the 'pressure drop/computability requirements' slot and finds that these requirements are satisfied. i.e., that there are two points in the same liquid, located one above the other, and not separated by a wall. The puller then multiplies the given values of ρ, g, and h and places the result in the 'pressure drop/result' slot.

As this example illustrates, the knowledge in specific quantity schemas and their pullers is domain-specific, here applying only to the quantity "pressure drop"

3.1.2. General knowledge

Of more central interest is FERMI's knowledge of general principles and methods that apply to a large variety of domains. This knowledge is encoded in two kinds of related schemas:

- General quantity schemas from which domain-specific quantities (like 'pressure drop') inherit knowledge

- Associated method schemas that encode related general problem-solving methods.

A general quantity schema contains pointers to one or more general methods. These pointers are inherited by all quantities related to that general quantity by any chain of *isa* links. Hence the method is general in that it applies to this entire class of quantities.

If a problem cannot be solved with pullers, FERMI checks the '/methods' slot of the schema for the quantity it wants to find. The corresponding 'quantity schema' may be connected by *isa* links to a 'general quantity' schema with a filled '/methods' slot containing a pointer to a general method. This pointer is inherited along the *isa* links to the 'quantity schema' of interest. If this is the case, FERMI tries this method to solve the problem.

For example, the current implementation of FERMI has a general quantity schema called 'decomposable quantity'. Its major slots are the following:

'/entity':   The decomposable quantity is a function of this specified entity. It can be decomposed with respect to any decomposition of this entity into component entities.

'/combination function':
                  This combination function specifies how the desired quantity can be found from the quantities associated with the individual component entities. (For example, this combination function might be scalar addition or some weighted average.)

The schemas for many specific quantities (including 'pressure drop', 'potential drop', and 'center of mass') are connected by *isa* chains to the schema 'decomposable quantity'. Correspondingly, all of these quantities inherit a pointer to the general method schema 'decomposition' which helps to decompose complex problems into simpler ones associated with component entities.

Specifically, the 'decomposition' schema specifies the following general method: Suppose that a quantity Q is, by virtue of a *isa* link, a 'decomposable quantity' with an '/entity' slot filled by a pointer to an entity E. Then the 'decomposition' schema contains a '/control-structure' slot filled by a piece of code which executes the following actions: (1) Decompose E so that each component entity E_i satisfies as many of 'E/computability requirements' as possible. (2) Construct schemas Q_i of the same '/type' as Q, associated with each of the subentities E_i. (3) Fill the 'Q_i/result' slot for each Q_i (using the associated pullers which can be applied because the computability requirements are satisfied). (4) Combine these results by using the 'Q/combination function' to yield the filler for the slot 'Q/result'.

schemas higher in the hierarchy of Figure 4. This inheritance, which causes each schema to function as if it contained all the slots and fillers of the schemas higher in the hierarchy, serves the following three major functions:

1. *A schema may inherit a puller.* For example, the 'difference quantity' schema in Figure 4 includes in its '/result' slot a puller that specifies how to find the drop in a quantity by subtracting its final value from its initial one. This puller is inherited by the 'pressure drop' schema so that it knows how to find the value of a pressure drop from a point A to a point B by subtracting the pressure at B from that at A. Encoding this knowledge in the more general 'difference quantity' schema makes it available not only to 'pressure drop', but also to other schemas such as 'potential drop'.

2. *A schema may inherit pointers to a general method.* For example, in Figure 4 the schema 'decomposable quantity' includes a pointer to the general 'path decomposition' method schema. Correspondingly, the 'pressure drop' schema inherits this knowledge and can thus also make use of this method. Because the knowledge of this general method is not directly encoded in the 'pressure drop' schema, it can be inherited not only by this schema but by many other specific quantities to which it applies.

3. *Schemas may inherit slots.* For example, 'pressure drop' (and all other quantity schemas) inherit a '/result' slot from the general 'quantity' schema. This use of inheritance helps to keep the system consistent, ensuring that related schemas have the same slots. A schema often inherits knowledge from more than one source. For example, the 'pressure drop' schema in Figure 4 inherits a puller from the 'difference quantity' schema and also a pointer to the general decomposition method from 'invariant sum over path'.

The preceding kinds of inheritance provide the following benefits:

1. *Inheritance simplifies coding.* General knowledge needs to be encoded only once and can then be used repeatedly by a large variety of more specific schemas.

2. *Inheritance makes it easy to apply old information in new ways.* For example, electric potential drop, like pressure drop, is a difference quantity and an invariant sum over path. The specific formulae for computing such a potential drop are, however, completely different. Nevertheless, the 'potential drop' schema in Figure 4 was easily added to the system after all the other schemas shown there had been completed. It provided only a small amount of domain-specific information about how to compute potential drops, but immediately inherited a large amount of essential

Figure 4:   FERMI's hierarchy of quantities
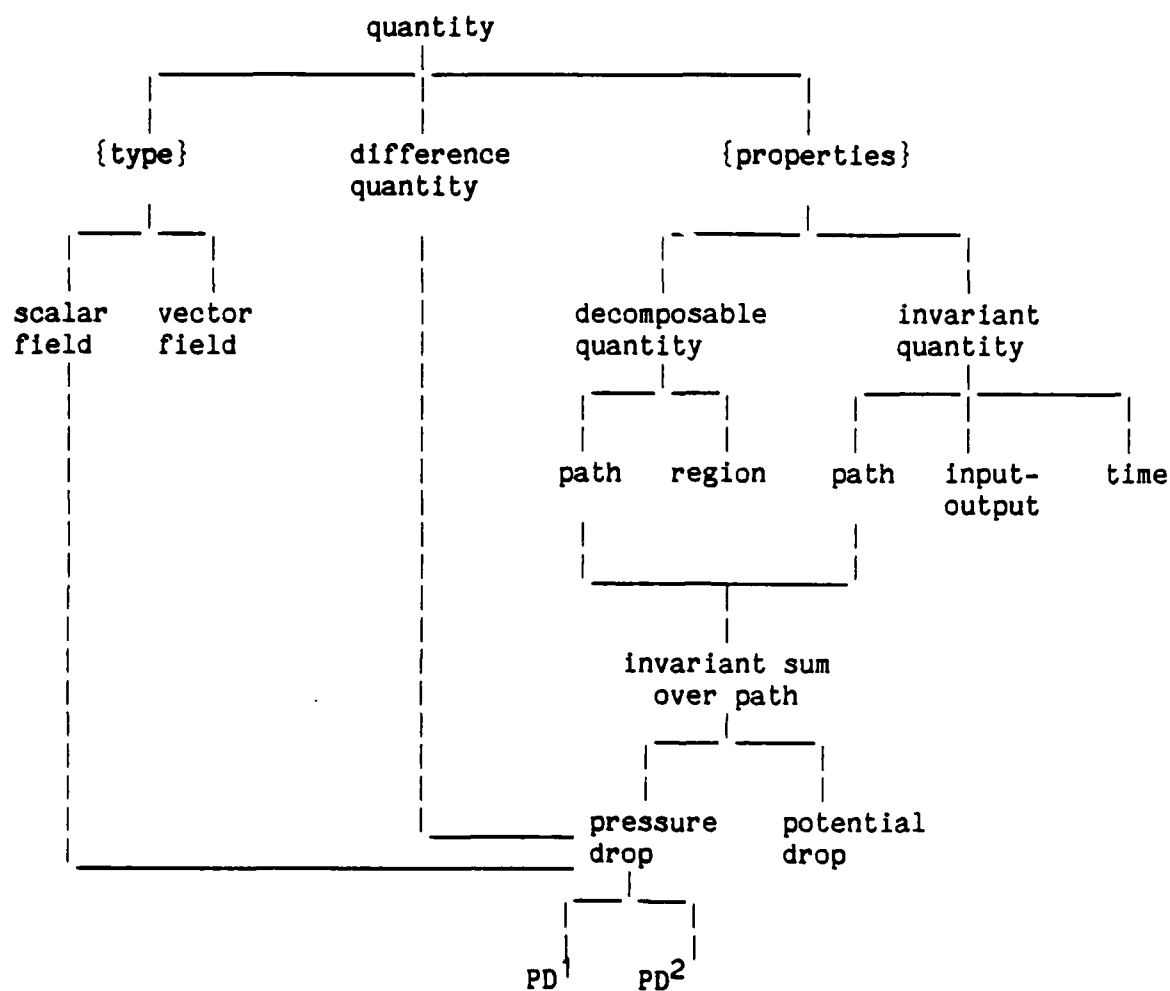
Direct application of the puller fails in this case because the weather vane is not rectangular in shape. Then FERMI tries to apply iterative decomposition because the 'center of mass/methods' slot contains a pointer to 'iterative region decomposition'. To do this, FERMI starts at an arbitrary corner of the weather vane, say corner A in Figure 3, and constructs a region satisfying the following requirements: (1) The region satisfies the 'center of mass/computability requirements' of being a rectangle. (2) The remaining region should be as small as possible, i.e., the chosen rectangle should be as large as possible. The chosen rectangular region is the rectangle ABCD indicated in Figure 3.

The 'region decomposition/done test' then determines that the computability requirements are satisfied by the three remaining rectangular regions. (Note that this '/done test', unlike that used for path decomposition, must be able to cope with multiple remaining regions.) FERMI then calculates the centers of mass for all four regions, and combines them using the 'center of mass/composition function' (average position weighted by mass).

3.3. Schema Hierarchies

The individual quantity and method schemas gain much of their power from their organization into schema hierarchies. Almost every schema (e.g., the 'pressure drop' or 'decomposition' schema) uses considerable knowledge not directly encoded in that schema. Instead, each schema is connected by *isa* links to other more general schemas. Knowledge encoded in schemas of high generality is then inherited by more specific schemas lower in the hierarchy. The following comments describe more fully the hierarchies encoding quantities and those encoding methods.

3.3.1. The quantity hierarchy

Figure 4 shows part of FERMI's hierarchy of quantity schemas. The previously described 'pressure drop' schema appears near the bottom of this hierarchy. It is connected by *isa* links (indicated by solid lines) to the more general schemas of 'difference quantity', 'invariant sum over path', and scalar field. Other *isa* links connect the 'pressure drop' schema indirectly to still more general schemas such as 'decomposable quantity' and 'quantity'. These general methods are employed only after pullers have failed to return an answer. In this way, FERMI exploits available specific methods before exploring more general ones.

Our earlier discussion of the 'pressure drop' schema in Table 1 included essentially only knowledge encoded directly in the 'pressure drop' schema itself. But the 'pressure drop' schema can, in addition, use all the knowledge inherited from

Figure 3:   A problem solved with iterative region decomposition
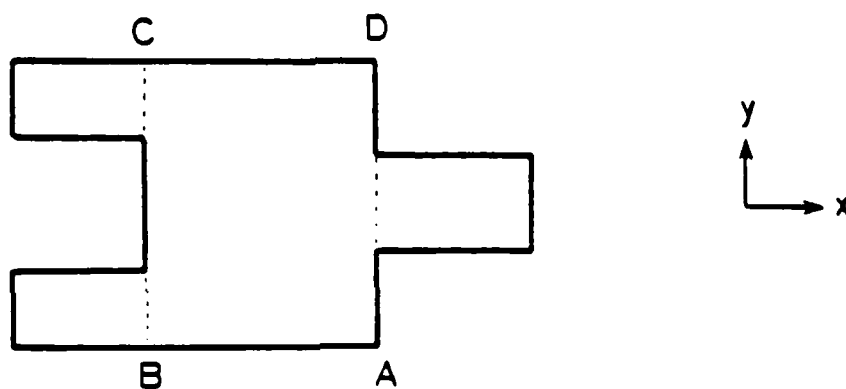
second general method schema associated with 'pressure drop'. This method tries to find a point Z satisfying the following conditions: (1) At least one of the new paths A-Z and Z-B violate fewer of the computability requirements than the original path A-B, and (2) neither path violates more computability requirements than the original path. The means for generating a path which does not violate a particular computability requirement is stored with the computability requirement itself. The result is the identification of the point Z shown in Figure 2. Note that the original path A-B violated all three computability requirements, i.e., that it should lie in a region of homogeneous liquid density, that it should not intersect any walls, and that it should be along a preferred direction (vertical or horizontal). By contrast, each of the new paths A-Z an Z-B violates only the single computability requirement that it should be along a preferred direction.

FERMI then finishes the problem by considering each of the new paths and decomposing these further, as indicated by the dotted lines in Figure 2 (by using either iterative or recursive path decomposition). Then the pressure drops along the four individual paths are computed and added to yield the answer to the problem.

Just as different control structures (iterative or recursive) can be used with the same type of entity (e.g., path), different types of entities can be used with the same control structure. Thus FERMI can use the iterative control structure either with paths or with regions. For example, the following problem is solved by iterative region decomposition.

Iterative region decomposition

The following problem concerns the calculation of the center of mass of a complexly shaped weather vane:

> A weather vane, having the shape shown in Figure 3, is made of sheet metal having a density of 2 gm/cm^2. What are the x and y coordinates of its center of mass where its supporting pivot should be placed?

As always, FERMI starts its work by trying direct domain-specific methods encoded as pullers. Here the 'center of mass' schema contains the following method, encoded in the puller of the '/result' slot and in the '/computability requirements' slot: If a rectangular object is aligned parallel to the x and y axes, with two parallel sides located at x_1 and x_2 and the other two parallel sides at y_1 and y_2, the center-of-mass coordinates x_c and y_c are found by computing the respective averages of these positions.

Figure 2: Figure illustrating a problem solved with recursive path decomposition

Table 2:   Examples of two method schemas

'Iterative path decomposition' schema

Control structure:
>   Iterative control structure.

Step generator: Iterative step generator.

Done test: Check on remaining step.

Entity type: Path.

Class membership {isa}:
>   Iterative decomposition.
>   Path decomposition.

'Path invariance' schema

Control structure:

Alternative getter:

Path generator:

Entity type:

Type of invariance:

Test to avoid duplication:

pointer to code for checking whether computability requirements are satisfied for a single complementary step.

The schema for the comparison-of-invariants method has, like all method schemas, a '/control structure' slot with code that executes when the method is called and that, in turn, calls code in other slots. The '/alternative getter' slot contains information about how to find or generate other situations to exploit the invariant properties of the quantity of interest.

For example, in the case of path invariance, these other situations are alternative paths between two points. As illustrated in Table 2, the schema contains then a '/path generator' slot accessing code that can generate a path either (1) parallel or perpendicular to a given direction, or (2) along pre-defined paths (e.g., along the wires of an electric circuit). The filler of '/entity type' indicates what kind of quantity (e.g., pressure drop) is invariant. The '/invariance relation' slot contains a pointer to the 'equation' schema holding the equation template for the particular kind of invariance. The code accessed from '/avoid duplication' prevents the same invariance equation from being generated redundantly for each of the quantities occuring in it. The way in which this is done may vary from one invariance method to another.

3.2.3. Examples of FERMI's functioning

We illustrate how FERMI uses its quantity and method schemas to solve two particular problems. The first of these requires recursive path decomposition and the second iterative region decomposition.

Recursive path decomposition

The following problem concerns the calculation of pressure drop within a liquid in a reentrant container:

What is the pressure drop from the point A to the point B in the water-filled container illustrated in Figure 2?

When FERMI tries to solve this problem, it successively tries to apply the available methods specified in the 'pressure drop' schema. In this case, the 'pressure drop/result' pullers fail because the conditions specified by '/computability requirements' are violated (i.e., intervening walls intersect the straight path from A to B). FERMI then tries to solve the problem by using 'iterative path decomposition'. This effort also fails for the following reasons: FERMI first constructs the path segments A-X and X-Y shown in Figure 2, thereby moving closer to B and satisfying all computability requirements. But when FERMI tries to iterate the process, considering the remaining path Y-B, it cannot find a path segment that begins at Y moves closer to B, and satisfies the 'pressure drop/computability requirements'.

At this point, FERMI tries the method of 'recursive path decomposition', the

reference to the following general method schemas (decomposition and comparison of invariants) incorporated in the present implementation of FERMI.

The schema for the decomposition method has slots specifying the kind of control structure used (iterative decomposition or recursive decomposition) and the kinds of entities (path or region) to which the decomposition is applied.

The control structure for *iterative* decomposition decomposes the original entity by generating a first component entity such that (1) the associated quantity is computable, and (2) the remaining complementary quantity poses a problem no more complex than the original one. The preceding process is then repeated (iterated) by being applied to the complementary quantity, and so forth, until the problem is solved or the solution is unachievable because no further decomposition is possible.

The control structure for *recursive* decomposition decomposes the original entity into two subentities such that (1) at least one poses a problem simpler than the original one, and (2) neither new problem is more complex than the original one. The method then repeats (recurses) the preceding process on each subproblem until the problem is solved or the solution is unachievable because a non-solvable subproblems cannot be further decomposed.

Both iterative and recursive decomposition involve step-by-step procedures with appropriate choices of subgoals, but with the following difference: In iterative decomposition the subgoals are chosen on the basis of *local* information (analogously to a hill-climbing strategy). In recursive decomposition the subgoals are chosen on the basis of *global* information about the entire problem (analogously to a divide-and-conquer strategy).

The implementation of both of these control structures requires tests for determining when a problem becomes more or less complex (i.e., less or more soluble). Such tests are incorporated in a 'more-soluble test' slot, referenced from the schema for the quantity of interest, and are applied in addition to the given computability requirements. (For example, in the case of the quantity 'pressure drop', this test specifies that a problem is more soluble if the pressure drop is to be found for a shorter path.)

> Table 2 illustrates the schema for the method of iterative path decomposition. Here the entity of interest is a path which can be decomposed into subpaths. The '/step generator' slot generates a single step (i.e., path segment) which (1) violates none of the computability requirements of the quantity being computed, (2) is as large as possible, and (3) produces a complementary problem more soluble than the original one. The '/done test' slot contains a

As illustrated in Table 1, each quantity schema includes a '/result' slot which is to be filled with the value of the quantity (here pressure drop). The puller associated with this slot first checks whether its '/computability requirements' are satisfied; if so it applies a domain-specific method to compute the quantity. (In Table 1, this method involves finding the indicated values for density, height, and gravitational constant.)

Table 1 also contains the slots '/point1' and 'point2' which contain pointers to schemas representing the two points between which the pressure drop is measured. The '/isa' slot contains one or more pointers to more general schemas with contents that are inherited by the current schema. Here the 'pressure drop' schema *isa* 'sum over path' schema; hence 'pressure drop' inherits all the contents of that schema, with results described later.

Pullers act recursively in the following sense. A puller may need the value of a slot that is currently empty and may therefore cause this slot's puller to act. This process continues until the system either encounters a slot with no puller, or succeeds in finding the filler of the originally desired slot. (For example, when the result of a 'pressure drop' schema is desired, its puller acts and requests a value for the vertical distance between the two specified points. If the '/result' slot for that schema is not filled, then it's puller acts to subtract the height-coordinates y of the two points. If these y coordinates are unknown, their pullers act.) This recursive puller architecture allows domain-specific knowledge to be encoded locally. In other words, each puller knows how to compute its own slot, but can interact with other knowledge anywhere in the system. [As illustrated later in discussing FERMI's performance , a puller, in fact, sets up an equation (e.g., $P = \rho g\ h$), and creates schemas for any quantity in that equation that doesn't already have one. A separate algebraic method computes values if possible.]

Pullers may return either numerical values or algebraic expressions. In the initial examples of this paper, the pullers return values. In a later section we consider a more complex example which illustrates FERMI's algebraic capabilities.

3.2.2. Method schemas

FERMI also uses schemas to encode knowledge of problem-solving methods. All method schemas have a '/control structure' slot which contains the code that executes when the method is called. To keep the control structure code simple, values and separable portions of code are stored in separate schemas, identified by pointers in slots of the main method schema. By separating knowledge into a variety of small procedures, FERMI can flexibly use the same code with differing control structures and applied to different entities. These comments can be illustrated with

Table 1:   'Pressure drop' schema.

Result:

Numerical value of pressure drop from point 1 to point 2.

<Return product of gravitational constant g, density of liquid at position of '/ point1', vertical distance between '/point1' and '/point2'>.

Computability requirements:

'/point 1' and '/point 2' aligned parallel or perpendicular to vertical direction.

These points in same liquid of uniform density.

These points not separated by a wall.

Lower point 1:

Upper point 2:

Type of quantity:

Scalar field.

Class membership {isa}:

Sum over path.

Difference quantity.

calls on the general method of path decomposition, inherited by the 'pressure drop/methods' slot from 'decomposable quantity'. This method seeks a set of path segments that (1) collectively make up a path from A to B, and (2) individually satisfy the 'pressure drop/computability requirements'. The result is the set of segments A-X, X-Y, and Y-B shown in Figure 1. This step uses general knowledge. FERMI then computes a pressure drop for each of these constructed segments, and then composes them (by addition) to obtain the originally desired pressure drop. This step uses both the domain-specific formula for pressure drop and a general method for combining quantities with a composition function (here addition). In other words, FERMI performs the following calculation for pressure drops indicated by the symbol P:

$$P_{AB} = P_{AX} + P_{AY}$$
$$= 0 + \rho_w \, g \, (2) + \rho_o \, g \, (1)$$
$$= 0 + (1.0) \, (980) \, (2) + (0.8) \, (980) \, (1)$$
$$= 2744 \text{ dyne/cm}^2$$

Here FERMI uses the known value $g = 980$ for the gravitational acceleration. (For simplicity, the present implementation of FERMI suppresses units, assuming that all quantities are specified in terms of the fundamental units of centimeter, gram, and second.)

In summary, FERMI's knowledge consists of hierarchically organized sets of general and domain-specific schemas for quantities and for methods. These schemas can be used either alone, or in interacting fashion, to solve problems that involve knowledge of varying degrees of generality.

3.1.4. Implementation language

The implementation of FERMI requires a language that can easily handle schema hierarchies and inheritance. We chose the schema-representation language SRL (Fox, 1979, Wright & Fox, 1983, 1983) in which isa links cause automatic inheritance of all the slots and their associated knowledge (fillers and pullers). This inheritance is transitive, i.e., it occurs along any chain of isa links. FERMI's power comes from using these schema hierarchies extensively to encode both declarative knowledge of principles and procedural knowledge of problem-solving methods.

The rest of this section describes in greater detail FERMI's quantity and method schemas, as well as their hierarchical organization.

3.2. FERMI's Schemas

3.2.1. Quantity schemas

A quantity schema consists of slots (which may or may not be filled). Each can store attributes of the quantity. For example, Table 1 shows the slots included in FERMI's schema for the quantity 'pressure drop'. Slots are listed along the left side, with fillers immediately following. Pullers are indicated by summaries (enclosed in angular brackets) describing their action.

Figure 1:   A pressure-drop problem requiring both general
and domain-specific knowledge
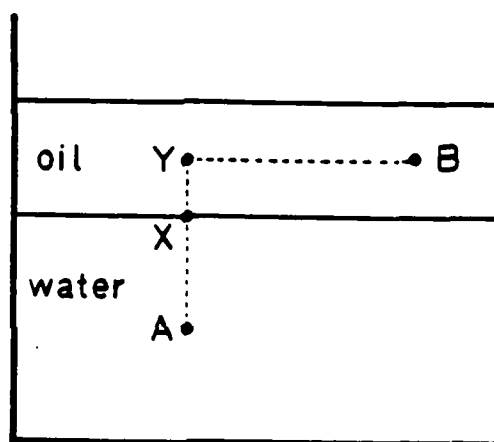
The structure used for 'decomposable quantity' is also used to implement 'invariant quantity'. A pointer to a general method for dealing with invariant quantities is included in a general quantity schema from which it is inherited by a variety of specific quantities. In this way the general method is encoded only once, but is accessible to all those quantities.

To illustrate the use of general methods, consider again the 'potential drop' schema. It inherits from 'decomposable quantity' a pointer to the decomposition method. 'Potential drop' also has slots and fillers specifying that it is decomposable over the entity 'path' and that its combination function is 'scalar addition'. (Both 'path' and 'scalar addition' are themselves also schemas.) This knowledge lets FERMI solve easily a simple problem like the following:

There is a 5 volt potential drop from point A to point B and a 10 volt potential drop from B to C. What then is the potential drop from A to C?

After the 'potential drop/result' puller fails, the 'known path decomposition' method decomposes the path from A to C into a path from A to B and one from B to C. Each of these segments trivially satisfies the 'potential drop/computability requirements' since the potential drops are already known for these path segments. FERMI then applies the 'potential drop/combination function' (i.e., scalar addition) to combine the individual potential drops (5 and 10 volts) and thus obtains the desired 'potential drop/result' of 15 volts.

The knowledge applied in this example is very general. The same knowledge would find the pressure drop from A to C by using knowledge about the pressure drops from A to B and from B to C. It would also find the center of mass of an object by applying its '/combination function' (average weighted by mass) to known centers of mass of component objects.

3.1.3. Combined knowledge application

FERMI can combine its domain-specific knowledge (stored in specific quantity schemas and their pullers) and its general knowledge (stored in general quantity schemas and their methods). If domain-specific knowledge alone fails to solve a problem, FERMI tries general methods. Usually, in contrast to the preceding examples, general methods alone do not completely solve the problem. Instead, the general methods require specific information which is supplied by the domain- specific quantity schemas and their pullers.

This process is illustrated by FERMI's solution to the following example.

A beaker is partly filled with water of density ρ_w = 1.0 gram/cm^3. A layer of oil, of density ρ_o = 0.8 gram/cm^3 floats on top, as shown in Figure 1. What is the pressure drop from a point A, located 2 cm below the water-oil interface, to another point B located 1 cm above the interface and 4 cm to the right of A?

This problem violates 'pressure drop/computability requirements' in two ways because the points are neither in the same liquid nor located vertically one above the other. Hence FERMI

information from the previously coded general schemas. As a result, very little new programming was required to allow FERMI to apply old knowledge to a new domain.
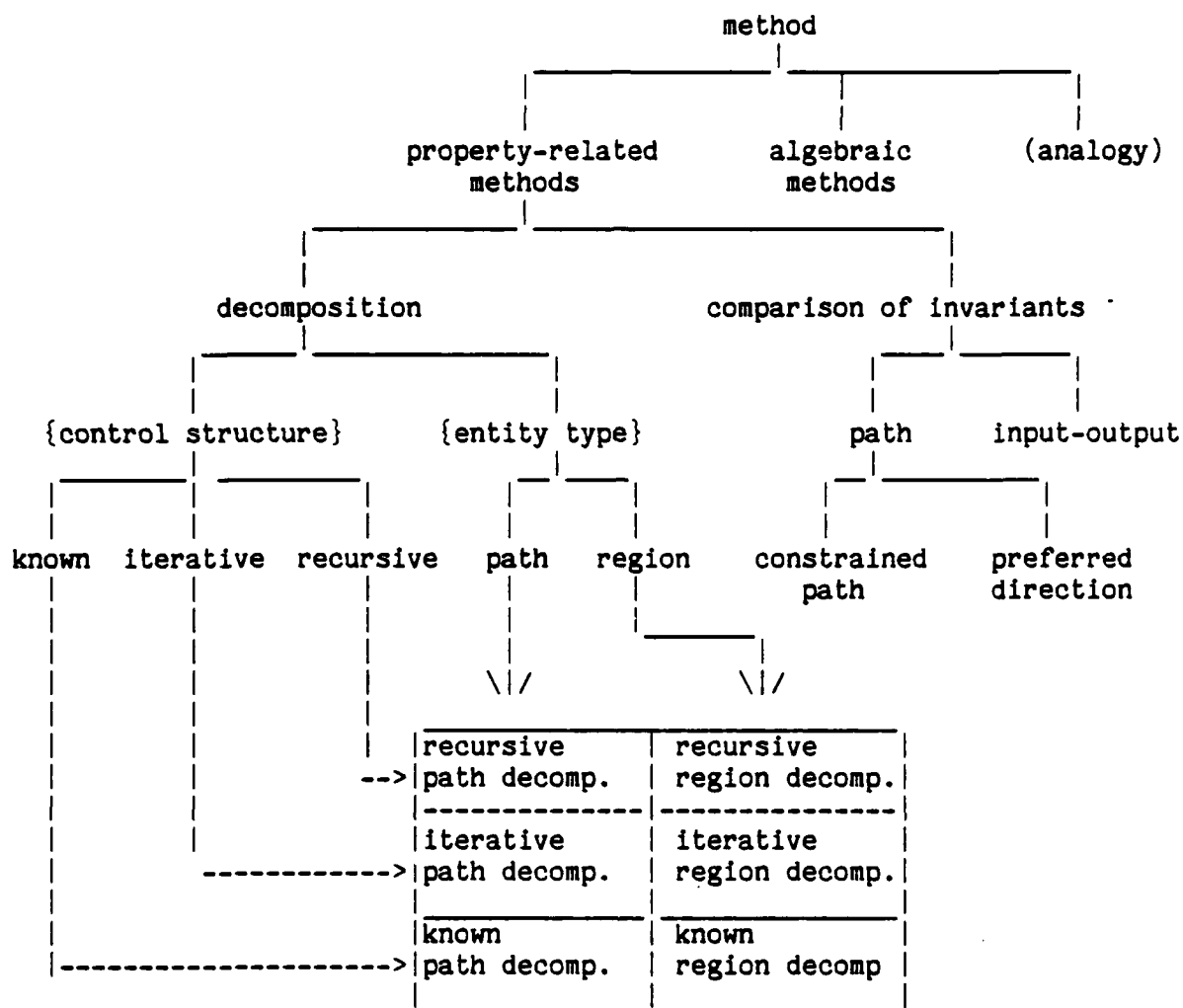
3. *Inheritance facilitates the encoding of specific problems.* For example, a specific quantity in a problem, such as a specific pressure drop PD^1, is simply encoded as a schema 'PD^1' that *isa* 'pressure drop.' Thus it automatically inherits all the knowledge of the quantity schema, including pointers to pullers and general methods. In addition, problem-specific knowledge (e.g. pointers to the specific points between which the particular pressure drop PD^1 is to be found) is encoded in the relevant slots of the particular PD^1 schema. Thus FERMI is provided with immediate access to both specific and general information.

4. *Inheritance provides an easy way to encode general principles.* Inheritance through *isa* links is the means whereby FERMI encodes what physical scientists call "principles". Thus the principle of conservation of energy, asserting that the energy of a system is an invariant quantity under certain conditions, is encoded by establishing an *isa* link between an appropriate energy schema and the 'invariant quantity' schema. This *isa* link gives the 'energy' schema all of the knowledge associated with an invariant quantity, including a pointer to all problem-solving methods exploiting invariance. Similarly, the *isa* chain between 'pressure drop' and 'invariant sum over path' in Figure 4 expresses the principle that pressure drops are path-invariant.

5. *Inheritance encourages consistency.* When a decision is made about how to encode certain knowledge (i.e., what slots to create etc.), that decision is implemented in the most general schema to which it is applicable. Inheritance then assures that that decision in consistently implemented for all of the specific schemas connected to that general schema.

3.3.2. The method hierarchy

Figure 5 summarizes the hierarchy that encodes FERMI's knowledge about general problem-solving methods. (Schemas in parentheses are planned, but not yet implemented.) The structure used to represent decomposable quantities is also used for invariant quantities. As in the quantity hierarchy, schemas higher in the hierarchy encode general knowledge that applies to several schemas lower in the hierarchy. By separating more general knowledge, it needs to be encoded only once and inheritance lets it apply in many specific situations.

The general method schema, at the top of the hierarchy in Figure 5, contains a '/control structure' slot which specifies code to be executed whenever the method is invoked. This slot is inherited by all other methods lower in the hierarchy. [For

Figure 5:   FERMI's hierarchy of major problem-solving methods

```
                                        method
                             _____|_____
                            |           |             |
                    property-related  algebraic    (analogy)
                        methods        methods
                  _____|_____
                 |                                   |
            decomposition                  comparison of invariants
         _____|_____                      _____|_____
        |                 |                     |           |
  {control structure}  {entity type}          path      input-output
     _____|_____          ___|___               _|_
    |     |     |        |       |             |     |
  known iterative recursive  path  region  constrained  preferred
    |     |     |        |       |            path       direction
    |     |     |        |       |
    |     |     |       \|/      \|/
    |     |     |    _____
    |     |     |   |recursive      | recursive          |
    |     |   -->|path decomp.    | region decomp.     |
    |     |      |----------------|--------------------|
    |     |      |iterative       | iterative          |
    |    ------------->|path decomp.    | region decomp.     |
    |            |_____|_____|
    |            |known           | known              |
    |----------------------->|path decomp.    | region decomp      |
                 |_____|_____|
```

example, the method of 'iterative path decomposition' inherits from the general 'decomposition schema' slots for '/step generator', '/done test', and '/entity type'. These slots are needed by all decomposition methods, but not by the other methods (such as invariance or analogy) indicated in Figure 5.] As in the case of the quantity hierarchy, this inheritance encourages consistency by ensuring that related schemas have the same slots.)

The method hierarchy allows different parts of a method to be encoded separately and used in a variety of contexts. For example, a decomposition method must include both control-structure information (about how to decompose a problem) and entity-type information (about what types of entities to decompose). Figure 5, and the following comments, indicate how these two ingredients of the decomposition method are encoded in two classes of descendants of the decomposition schema.

The method schemas contain filled '/step generator' slots with contents corresponding either to recursive, iterative or known decomposition methods. In the 'iterative decomposition' schema, this slot is filled with pointers to code that specifies how to construct a first solvable problem. and how then to iterate this process to generate a sequence of such solvable problems. In the 'recursive decomposition' schema, this slot contains pointers to code that specifies how to subdivide a problem into two subproblems, and how then recursively to construct more such subproblems. In the 'known path decomposition' the code simply identifies subproblems already present in the problem representation. The entity-type schemas contain slots with fillers specifying whether the entity of interest is a path or a region. They also contain slots with pointers to schemas providing knowledge of the details of decomposition specific either to paths or to regions.

The structure of the hierarchy, involving the invariance method for comparing invariants, is quite similar. The general 'comparison of invariants' schema contains a control structure specifying how to generate equations exploiting the invariance property of a quantity. As indicated in Figure 5, this control structure is then inherited by the more specific invariance methods lower in the hierarchy (e.g., by invariance with respect to changed paths, or invariance with respect to changes from input to output current at a node in an electric circuit). The schemas for 'path invariance' or 'input-output invariance' have different contents for the '/alternative getter' slot which generates the changed entity to be considered. In the case of preferred-direction invariance, the '/alternative getter' uses the contents of a '/path generator' slot that generates paths parallel or perpendicular to a preferred direction (e.g., vertical or horizontal paths); in the case of constrained invariance, the '/alternative getter' generates paths among those specified in a problem (e.g., along the wires in a given

electric circuit).

The preceding separation of various kinds of knowledge about methods has the following advantages: (1) Pertinent knowledge is encoded with less repetition. For example, FERMI can combine any 'control structure' schema with any 'entity type' schema, yielding four separate schemas for decomposition. Each part of the method is used in two of these methods, but needs to be encoded only once. (2) It is easily possible to use old knowledge in new ways. For example, if a new 'entity type' schema is added, FERMI can automatically use it with any of its 'control structure' schemas; similarly, if a new 'control structure' is added, FERMI can immediately use it with any 'entity type'.

3.3.3. Other schemas in FERMI's hierarchical knowledge

All of FERMI's knowledge is encoded in the form of hierarchically organized schemas. Thus the hierarchy of quantity schemas in Figure 4 is only a part of the more encompassing hierarchy of entity schemas illustrated in Figure 6. (This hierarchy includes various kinds of objects, in addition to the previously discussed quantities describing such objects.) Similarly, the hierarchy of method schemas in Figure 5 is only a part of the more encompassing hierarchy of action schemas in Figure 7. (This hierarchy includes various simple actions, in addition to the more complex methods described previously.)

For example, the 'object' schemas in Figure 6 contain slots for physical properties like mass and density, They also contain slots for geometric properties like height and width, and pointers to point schemas that specify the physical boundaries of the object. (Not all these slots need to be filled; e.g., the slots for mass or density in the case of circuit elements).

As mentioned previously, quantity schemas may contain pointers to various simple action schemas, e.g., to tests for computability requirements or to tests for better solubility.

The entity schemas in Figure 6 are simpler and inherit less complex knowledge than the general quantity schemas discussed previously. Similarly, the more primitive action schemas in Figure 7 are simpler and inherit less complex knowledge than the general method schemas discussed previously. Nevertheless, the hierarchical structure of all this knowledge retains the previously mentioned advantages: Knowledge relevant to any group of schemas needs to be encoded only once in a higher-level schema.

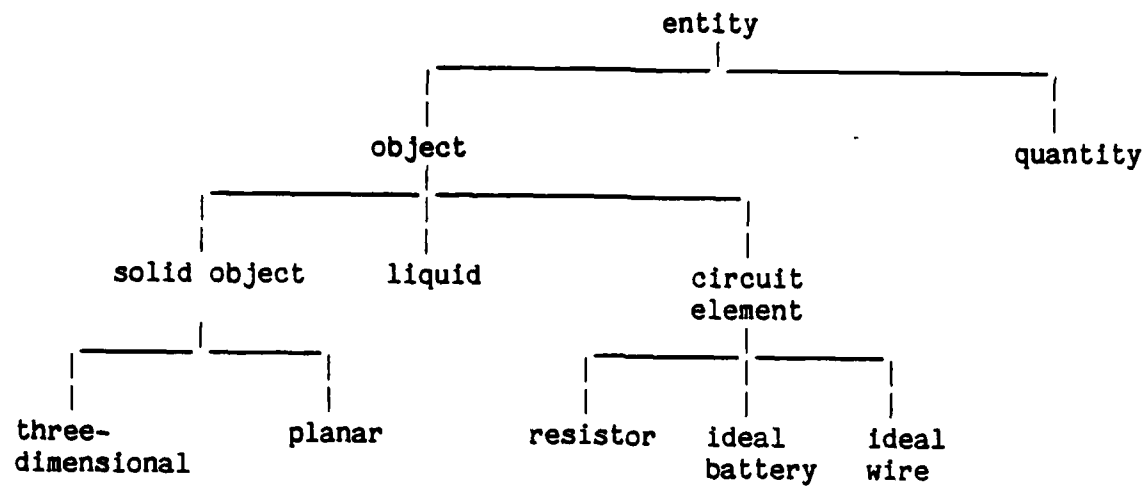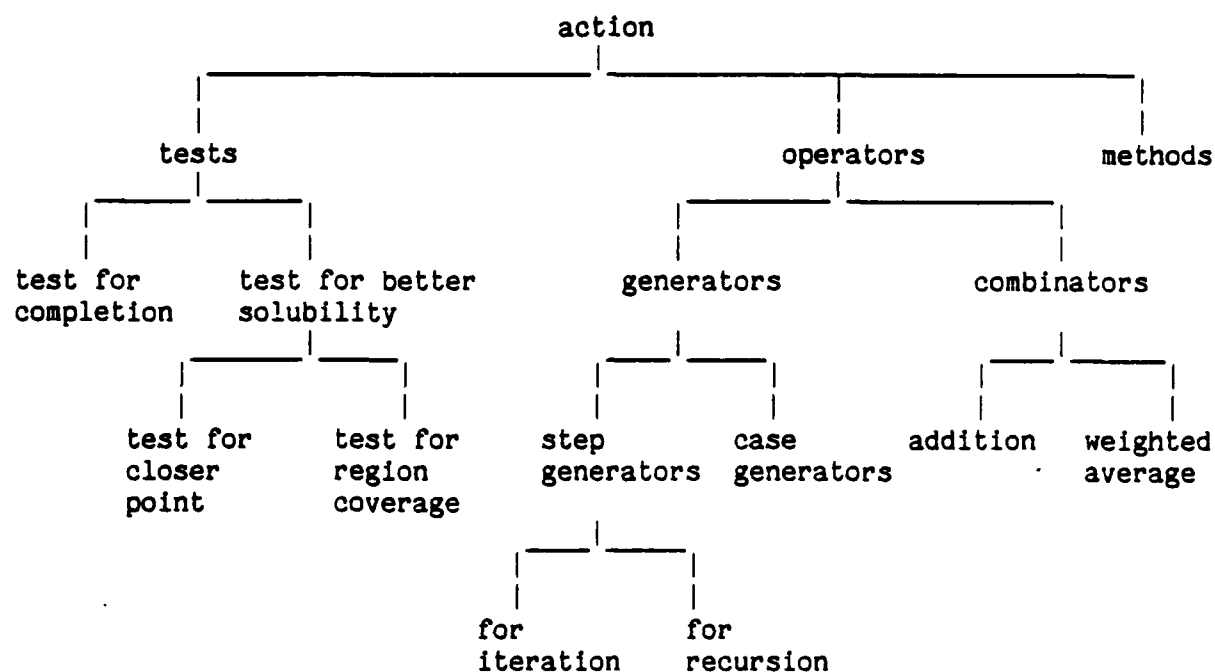**Figure 6:** FERMI's hierarchy of entitles (including the previously described hierarchy of quantities

```
                                  entity
                        _____|_____
                       |                               |
                     object                         quantity
              _____|_____
             |                           |
        solid object     liquid      circuit
                                     element
          ____|____              _____|_____
         |         |            |      |      |
       three-    planar      resistor ideal  ideal
       dimensional                   battery wire
```

Figure 7: FERMI's hierarchy of actions (including the previously described hierarchy of methods

```
                                        action
                                          |
        ┌─────────────────────────────────┼─────────────────────────┐
        |                                 |                          |
      tests                           operators                  methods
        |                                 |
   ┌────┴────┐                   ┌─────────┴─────────┐
   |         |                   |                   |
 test for  test for better   generators          combinators
 completion solubility           |                   |
        ┌──────┴──────┐      ┌────┴────┐        ┌─────┴─────┐
        |             |      |         |        |           |
     test for     test for  step      case    addition   weighted
     closer       region    generators generators          average
     point        coverage      |
                            ┌────┴────┐
                            |         |
                           for       for
                        iteration  recursion
```

4. Performance

This section discusses FERMI's performance by presenting traces of its work on three problems.   The first trace concerns a simple fluid statics problem solved by domain-specific knowledge alone (i.e., by pullers). This trace illustrates the format and content of FERMI's problem solutions. The second trace solves a circuit problem and uses both domain-specific knowledge and the general method of decomposition; this trace also introduces FERMI's use of algebra. The third trace solves a more complex circuit problem, the most difficult problem FERMI has solved.   This trace illustrates FERMI's combined use of decomposition and invariance, together with algebraic reasoning about sets of equations.

4.1. Use of Domain-Specific Knowledge

Figure 8 shows FERMI's solution to a problem requiring only domain-specific knowledge. FERMI's traces are organized as nested sets of goals (each labeled by G followed by a number) and corresponding results (labeled by R with the same number).   Vertical lines join corresponding goals and results.   In Figure 8 the first goal G1 is to find the pressure drop (called *pressure-drop-1*).   FERMI first checks whether its value is available in the 'pressure-drop-1/result' slot ("Lookup" in the trace).   Finding that the value is not already available, FERMI then tries to apply domain-specific knowledge (pullers).   This is done in two steps.   First, the 'pressure drop' pullers yield two expressions (R2) for pressure-drop-1.   [Expressions are written with an operator (+, -, *, /) preceding the symbols on which it operates for example (* density-1 g rel-height-1) denotes the product of these three quantities.]   Then goal G3 yields the numerical value (R3) produced by substituting known values into one of these expression.

In more detail, goal G2, to apply pullers causes both of the pullers associated with 'pressure drop' to be applied.   The first (inherited from the 'drop' schema) says that a pressure drop is equal to the difference of the initial minus the final pressure. This puller has no computability requirements--the relation applies to any drop FERMI therefore builds expression-1 relating the desired pressure-drop-1 to the absolute pressures (pressure-1 and pressure-2) at the initial and final points.   FERMI then applies the second puller associated with the 'pressure drop' schema.   Its computability requirements are satisfied--the two points A and B lie in the same container, in a region of homogeneous density, and are vertically aligned.   This puller therefore returns *expression-2*, equal to the product of density, gravitational acceleration, and relative height.

The pullers produce result R2, including *expression-1* and *expression-2*.   Goal

Figure 8:   Trace of FERMI's work on a problem
requiring only domain-specific knowledge.

Problem: What is the pressure drop in a beaker of water (of density 1 gm/cm^3) from a point A located 1 cm above the bottom to a point B located above A and 3 cm above the bottom?

Panel 1:   Main Steps of the Solution

G1: pressure-drop-1

Lookup: empty

G2: Apply-pullers

Apply puller: (- initial-pressure final-pressure)

| Computability-requirements: satisfied

pressure-drop-1 = expression-1 [- pressure-1 pressure-2]

Apply puller: (* density g relative-height)

| Computability-requirements: satisfied

pressure-drop-1 = expression-2 [* density-1 g rel-height-1]

R2: pressure-drop-1 = expression-1, = expression-2

G3: evaluate-expressions-for-pressure-drop-1

   OR { expression-1 expression-2 }

    G4: Evaluate expression-1: (- pressure-1 pressure-2)

      AND { pressure-1 pressure-2 }

       G5: pressure-1

       |Lookup: empty

       G6: Apply-pullers

       |none

      R6: fails

     R5: fails

   R4: fails

   G7: Evaluate expression-2: (* density-1 g rel-height-1)

      AND { density-1 g rel-height-1 }

      G8: density-1

      |Lookup: 1.0

      R8: density-1 = 1.0

      G9: g

      |Lookup: 9.8

      R9: g = 9.8

      G10: rel-height-1

      | ⋮

      | *This part of trace elaborated in Panel 2*

      R10: rel-height-1 = 2.0

   R7: expression-2 = 19.6

R3: evaluate-expressions-for-pressure-drop-1:   19.6

R1: pressure-drop-1 = 19.6

Panel 2
Finding Relative Height

(Figure 8 continued)

```
G10: rel-height-1
 |Lookup: empty
 |G11: Apply-pullers
 | |Apply puller: (- height-2 height-1)
 | | |Computability-requirements: satisfied
 | |rel-height-1 = expression-3 [- y2 y1]
 |R11: rel-height-1 = (expression-3)
 |G12: evaluate-expressions for rel-height-1
 |     OR { expression-3 }
 |G13: Evaluate expression-3: (- y2 y1)
 | |    AND { y2 y1 }
 | |    G14: y2
 | |    |Lookup: 3.0
 | |    R14: 3.0
 | |    G15: y2
 | |    |Lookup: 1.0
 | |     R15: 1.0
 |R13: expression-3 = 2.0
 |R12:  Evaluate-expressions-for-rel-height-1:  2.0
R10: rel-height-1 = 2.0
```

G3 is then to find the desired pressure-drop-1 by evaluating either of these two expressions. G3 therefore produces the subgoal (labeled *OR*) aiming to evaluate *either* of the expressions enclosed in curly brackets. Goal G4, to evaluate *expression-1*, is unsuccessful. The reason is that FERMI currently has no pullers associated with absolute pressures. Therefore no values can be found for the pressures 'pressure-1' and 'pressure-2' at the points *A* and *B*. Goal G7 (to evaluate *expression-2*) can be achieved by attaining the subgoal (indicated by *AND*) of finding jointly the three quantities indicated in the curly brackets. The three elements of the *AND* goal become the individual goals G8, G9, and G10. Density-1 (G8) and g (G9) are found through a simple *lookup*, i.e., their values are already available in the '/result' slots of their schemas. The subgoals of (G10), Rel-height-1, are given in Panel 2 of Table 8. Rel-height-1 requires a puller to produce expression-3 relating rel-height-1 to the absolute heights y-1 and y-2 of points *A* and *B* (G11-R11). Then this expression is evaluated (G12-R12) to yield the value of rel-height-1 (R10).

With values for density, for g, and for relative height, the value of *expression-2* can be found (R7). In turn, this satisfies goals G3 (to evaluate an expression for *pressure-drop-1* and G1 to find the value of *pressure-drop-1*.)

4.2. Use of Domain-specific and General Knowledge

Figure 9 shows a trace of FERMI's work on a circuit problem that requires the general method of path decomposition as well as domain-specific knowledge. This trace is divided into two panels, with Panel 1 showing the main steps of the solution, and panel 2 showing details of one part of the trace. Very generally, the top goal (G1: find *potential-drop-0* from *a* to *b* in Figure 9) is achieved in three main steps. Pullers are tried (G2-R2), but do not succeed. FERMI then uses methods to generate an expression for the desired potential drop (G3-R3), and successfully evaluates this expression (G4-R4).

In more detail, the puller for finding potential drop fails because it applies only to a path that is a single circuit component. [As in Figure 8, the puller inherited from 'drop/methods' produces an expression (- potential-1 potential-2) which cannot be evaluated because there are not pullers for absolute potential. The action of this 'drop' puller is omitted from this and subsequent traces.] FERMI then identifies its applicable general methods. These methods include: (1) simple known path decomposition (kpd) with which FERMI decomposes a path into already known parts, (2) iterative path decomposition (ipd), (3) recursive path decomposition (rpd), and (4) path invariance. The pointer to these methods is inherited by potential drop from the general quantity schema 'quantity decomposable over path'.

Figure 9:   Trace of FERMI's work on a problem requiring
domain-specific and general knowledge.

Problem: A circuit contains a battery, with an emf of 6 volts, and a resistor
of resistance 1 ohm.   A current of 3 amperes flows through the resistor from a to b
as shown in Figure 8.   What is the potential drop from the point a to the point b?

Panel 1
Main Steps of the Solution

G1: potential-drop-0 [from a to b]
Lookup: empty
G2: Apply-pullers
| Apply puller: (either (* current resistance) or (- emf))
| | path-lies-in-same-component not satisfied
| fails
R2: fails
G3: Apply methods { kpd ipd rpd path-invariance }
| Apply method: kpd
| fails
| Apply method: ipd
| potential-drop-0 = expression-1
| {+ potential-drop-1 potential-drop-2]
R3: potential-drop-0 = expression-1
G4: evaluate-expressions-for-potential-drop-0
| | OR {   expression-1 }
| | G5: evaluate expression-1: (+ potential-drop-1 potential-drop-2)
| | | AND { potential-drop-1 potential-drop-2 }
| | | G6: potential-drop-1
| | | | •
| | | | •
| | | | •
| | | | This part of trace
| | | | elaborated in panel 2.
| | | R6: potential-drop-1 = - 6.0
| | | G11: potential-drop-2
| | | |
| | | R11: potential-drop-2 = 3.0
| | R5: expression-1 = -3.0
| **R4: evaluate-expressions-for-potential-drop-0:   -3.0**
R1: potential-drop-0 = -3.0

Panel 2
Finding Potential-Drop-1 and Potential-Drop-2

(Figure 9 continued)

G6: potential-drop-1
| Lookup: empty
| G7: Apply-pullers
| |Apply puller: (* current resistance) or (- emf))
| | |Computability-requirements: satisfied
| |potential-drop-1 = expression-2 [- emf-1]
| R7: potential-drop-1 = (expression-2)
| G8: evaluate-expressions-for-potential-drop-1
| | | OR { expression-2 }
| | | G9: evaluate expression-2: (- emf-1)
| | | | AND { emf-1 }
| | | | G10: emf-1
| | | | |Lookup: 6.0
| | | | R10: emf-1 = 6.0
| | | R9: expression-2 = - 6.0
| R8: evaluate-expressions-for-potential-drop-1:   - 6.0
R6: potential-drop-1 = - 6.0
G11: potential-drop-2
| Lookup: empty
| G12: Apply-pullers
| |Apply puller: (* current resistance) or (- emf))
| | Computability-requirements: satisfied
| |potential-drop-2 = expression-3 [* current-1 resistance-1]
| R12: potential-drop-2 = (expression-3)
| G13: evaluate-expressions-for-potential-drop-2
| | | OR { expression-3 }
| | | G14: evaluate expression-3: (* current-1 resistance-1)
| | | | AND { current-1 resistance-1 }
| | | | G15: current-1
| | | | |Lookup: 1.0
| | | | R15: current-1 = 1.0
| | | | G16: resistance-1
| | | | |Lookup: 3.0
| | | | R16: resistance-1 = 3.0
| | | R14: expression-3 = 3.0
| R13: evaluate-expressions-for-potential-drop-2:   3.0
R11: potential-drop-2 = 3.0

Figure 10:   Battery and resistor in series.

E (6 volt)

R (1 ohm)

a   x

I (3 amp)   b
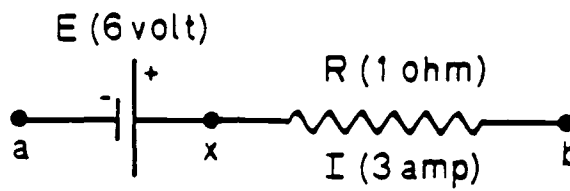
The first of these methods, known path decomposition (kpd), fails because there are no pre-specified path components. The second, iterative path decomposition (ipd), produces an expression, called *expression-1*, that relates the desired potential-drop-0 to the sum of potential-drop-1 from *a* to *x* across the battery, and potential-drop-2 from *x* to *b* across the resistor. FERMI chooses these path components because each is a single circuit element and so satisfies the computability requirements of the puller for potential drop.

Having succeeded in producing (R3), an algebraic expression for the desired potential drop, the next goal, G4, is to evaluate all available expressions for the desired potential-drop-0. The *OR* goal under G4 indicates that FERMI would try to evaluate any one of a set of available expressions, although here there is only one. Evaluating expression-1 (G5) requires finding values for *both* potential-drop-1 and potential-drop-2 (indicated by the *AND* goal below G5 and by the subsequent goals G6 and G11). Goals G6 and G11 both succeed because in both cases a puller associated with 'potential-drop' yields an expression that can be evaluated to find the desired potential drop. (These details are given in Panel 2.) The values (potential-drop-1 = - 6.0 and potential-drop-2 = 3.0) are used to evaluate expression-1 (R5) which is the value of the originally desired potential-drop-0 (R1).

Panel 2 of Figure 9 shows details of how FERMI achieves goals G6 and G11 and finds values for potential-drop-1 and potential-drop-2. In both cases the computability requirements for the 'potential-drop' puller are satisfied. In each case the puller yields an expression: expression-2 = (- emf) for potential-drop-1 across the battery and expression-3 = (* current resistance) for potential-drop-2 across the resistor. Then these expressions are evaluated (G8-R8 and G13-R13) to yield values for the two potential drops.

4.3. Use of Several Kinds of Knowledge

Figure 11 shows a more complex circuit problem and FERMI's solution to it. Again the trace is divided into panels, with panel 1 showing the main goals and results, and subsequent panels giving more details.

In panel 1, the desired current, called current-0 in the trace and I_0 in Figure 12, is found in three steps. (G2-R2) "Lookup" and use of pullers fail, the latter because the wire through which I_0 flows in neither an ideal battery nor an ideal resistor with non-zero resistance. In (G3-R3) FERMI applies methods, considering input-output invariance, the only method applicable to currents. FERMI applies this method first to a node involving three wires (and hence three currents)--in this case

Figure 13:   Simulated trace showing use of constraints based in invariance

Problem: Two resistors of resistance R1 = 3 ohm and R2 = 2 ohm are connected in parallel as shown in Figure 12.   The battery has an emf of 6 volts.   What is the current I_0 flowing through the wire from a to b?

```
G0:   I_0
   │  G1:   Constraint equations
   │     │  OR  {Input-output invariance,
   │     │        Path invariance}
   │     │     G2:   Input-output invariance
   │     │     R2:   {I_0 = I_1 + I_2}
   │     │     G3:   Path invariance
   │     │     R3:   {V1 = V0, V2 = V0}
   │  R1:   {I_0 = I_1 + I_2,
   │           V1 = V0, V2 = V0}
   │  G4:   Pullers
   │     │  G5:   V1
   │     │  R5:   V1 = R1 I_1 = 2 I_1
   │     │  G6:   V2
   │     │  R6:   V2 = R2 I_2 = 3 I_2
   │     │  G7:   V0
   │     │  R7:   V0 = 6
   │  R4:   {V1 = 2 I_1, V2 = 3 I_2,
   │           V0 = 6}
   │  G8:   solve equations
   │  R8:   I_0 = 5
R0:   5
```

5.3.2. *Methods of constraint satisfaction.*

To exploit the power achievable by expressing its general principles as various kinds of constraints, FERMI would need an associated general method of constraint satisfaction. As discussed elsewhere (Reif & Heller, 1982), such a method involves the following two major steps: (1) Finding enough constraints, applicable to a particular problem, so that only one solution is consistent with all of these constraints. (2) Finding a solution by trying to satisfy all of these constraints.

To implement the first of these steps, FERMI would approach a problem with the specific goal of finding applicable *constraints* involving the quantities of interest in the problem. (To do this, FERMI can apply the constraints expressing general principles and instantiate them in the particular problem situation). This goal is different from that, pursued in the current implementation. where FERMI solves a problem by backward chaining (proceeding in linear sequence by trying to find a particular *quantity* of interest, then pursuing the subgoal of finding further quantities relevant to it, and continuing in this way with successive subgoals until everything is reduced to known information). In other words, FERMI would shift its goal from trying to find particular *quantities* to that of finding *constraints* relating such quantities.

For example, Table 13 shows a possible constraint-based solution to the circuit problem already presented in Figures 11 and 12. The earlier solution was organized around finding values for the individual quantities I_1 and I_2 (currents through the resistors) in order to combine them to find the desired quantity I_0. The solution in Figure 11 is organized around three constraint equations arising from applying input-output invariance to node b [See Figure 12) and path invariance to alternate paths between nodes b and c (the path through R_1 and the path through R_2)(the path through R_1, and the path through R_2)]

The first phase of the problem, aiming to satisfy goal G1, produces three constraint equations. The second phase (G4) produces local equations relating quantities in the constraint equations to quantities given in the problem. The third phase (G8) solves these equations to find the answer. The only difficulty in the first phase is selecting which invariance to apply. Conservation of momentum, for example, would not contribute to this problem solution. FERMI must have some selection mechanism in order to reject such irrelevancies. We believe that suitable selection of invariance methods can be achieved in the following way: consider just hose invariance principles listed in the '/method' slots of schemas in the problem or schemas directly related to these quantities. In this problem, for example, we use invariance principles appearing in the '/method' slots of schemas current, resistance, and emf which are mentioned in the problem. We also consider path invariance, appearing in the schema for potential drop, a quantity directly related to current, resistance, and emf.

To implement the second major phase. that concerned with finding a solution from specified problem-specific constraints expressed as equations, FERMI would merely need sufficient algebraic capabilities to manipulate symbolic equations and to solve sets of simultaneous equations involving several unknown quantities. FERMI already possesses most of these capabilities. although they could be further refined.

By solving problems with such a method of constraint satisfaction, FERMI's goal

5.3. Additions to FERMI's General Knowledge

5.3.1. Principles formulated as constraints.

The general principles incorporated in FERMI's knowledge specify the properties of certain quantities. Such properties can be regarded as constraints on the possible values of such quantities. Usually these constraints can be expressed in the form of equations (although some may be expressed as inequalities).

From this point of view, FERMI's two general principles of decomposability and invariance can be viewed as expressing particular kinds of constraints. The decomposability principle asserts that some quantity Q (e.g., pressure drop), describing some entity E (e.g., path), can be obtained from some specified function of the quantities Q_i associated with all the component entities of E. (In the particular case of additive decomposability, the decomposability principle can then be expressed by the constraint $Q = \Sigma Q_i$.) Similarly, the invariance principle asserts that some quantity Q remains unchanged. Hence this principle can be expressed by the constraint that $Q_i = Q_j$ for any two values Q_i and Q_j of the quantity.

FERMI's capabilities could be appreciably increased by augmenting FERMI's knowledge with some additional general principles specifying other important kinds of constraints. One such additional important general principle would be a "fixed-value constraint" specifying that some quantity has some particular fixed value. (Often, such a constraint can be expressed in the form $Q = 0$, where the fixed value is zero.)

For example, in the domain of mechanics, an object remains in equilibrium (without accelerating) if the total force F exerted on the object by all other interacting objects satisfies the principle expressed by the constraint that $F = 0$. As another example, in the domain of electricity, an electrically neutral object must have a total charge Q which satisfies at all times the charge-conservation principle that $Q = 0$.

Note that the invariance principle can also be expressed as a fixed-value constraint on differences of the relevant quantity. In symbols, the invariance constraint that $Q_i = Q_j$ is equivalent to the statement that the difference $\Delta Q = Q_i - Q_j$ satisfies the constraint $\Delta Q = 0$.

these domains.

5.2.2. Domain principles subsumed by invariance

In its present implementation, FERMI has used its general invariance principle (and associated method for comparing invariants) only to a very limited extent. It has used invariance under choice of path for the calculation of pressure drops in a liquid, or for calculations of potential drops in an electric circuit. It has also used invariance between the total current flowing into the junction point of such a circuit and the total current flowing out of such a junction point. However, many of the most interesting and important applications of invariance have remained unexploited. By exploiting the invariance principle more fully, FERMI should be able to apply this principle to subsume more specific principles of great importance and to apply these to solve problems in a great variety of domains. The following are some important cases:

- *Invariance under changes of time.* A very important kind of invariance is that asserting that the value of some particular quantity remains unchanged in the course of time. (Such an assertion is a "conservation principle" asserting that the corresponding quantity remains invariant in the course of time.) For example, in mechanics the *momentum* of a system remains invariant in the course of time under certain conditions, i.e., the "principle of conservation of momentum" is a special kind of invariance. Similar statements are true about the quantity *angular momentum.* The *energy* of a system also remains invariant in the course of time in many cases, with the result that the principle of "conservation of energy" is one of the most important principles in physics, chemistry, and biology.

- *Invariance under spatial changes.* Invariance under particular geometrical spatial transformations leads to important "symmetry properties". (For example, a square remains invariant if rotated through a 90° angle or 180° angle about a perpendicular axis through its center. It also remains invariant if reflected about one of its diagonals.) The exploitation of such symmetry properties can often greatly simplify many computations since the values of many quantities describing symmetrical objects are often apparent from symmetry properties alone.

- *Relation between decomposability and invariance.* Decomposability may imply invariance under certain conditions. Indeed, if a particular quantity can be decomposed in several different ways, then this quantity is invariant under these different modes of decomposition. (For example, if the pressure drop between two points can be decomposed into a sum of pressure drops along successive segments of *any* path joining these points, then the pressure drop is invariant under changes of path.)

problems.

- *Mechanics.* The *mass* of an object can be decomposed into the sum of the masses of constituent parts of this object. As already discussed, the position of the *center of mass* of an object can be decomposed (by a weighted average) into the positions of the centers of mass of the constituent parts of this object. Most important, the *force*, exerted on an object by other interacting objects, can be decomposed into the vector sum of the forces exerted on this object by all the other objects interacting separately. This last principle is the "superposition principle" of central importance in Newton's laws of classical mechanics.

- *Electricity and magnetism.* The *electric field* produced at any point by any number of charged particles can be decomposed into the vector sum of the electric fields produced by these charged particles separately. A similar principle applies to the *magnetic field* and to *potentials* associated with these fields.

- *Heat and thermodynamics.* The *internal energy* of any system can be decomposed into the sum of the internal energies of the constituent parts of this system. Similarly, the *entropy* of any system in equilibrium can be decomposed into the sum of the entropies of all the constituent parts of this system.

- *Chemistry.* The *molecular weight* of a molecule can be decomposed into the sum of the molecular weights of all the individual atoms in the molecule. As a more complex example, the *reaction rate* of several reacting chemical species present jointly can be decomposed into a product involving the concentrations of these species.

- *Waves.* This rich domain encompasses all kinds of waves (including water waves, sound waves, radio waves, light waves, etc.) and, properly interpreted, includes even quantum mechanics. Here the *wave disturbance*, due to several waves present simultaneously, can be decomposed into the sum of the disturbances due to the individual waves present separately. (This is the "superposition principle" of central importance for all waves and is the basis for all problems involving interference phenomena.) In addition, any wave disturbance (as well as any other time-dependent quantity) can always be decomposed into a sum of sinuosoidal waves of different frequencies. (This is Fourier's famous principle.)

The preceding list demonstrates the very wide applicability of the decomposition principle presently incorporated in FERMI's general knowledge. Hence FERMI's general knowledge and associated methods can be readily extended to a very large range of domains by the addition of limited amounts of more specific knowledge about

important in limited domains. Because FERMI's knowledge is hierarchically organized, knowledge of these very general principles, and of their associated methods, is automatically inherited so as to be directly applicable to these more domain-specific principles.

FERMI's present general knowledge about decomposability and invariance, even if extended no further, subsumes a much larger number of domain-specific principles of great importance -- with only a small addition of specific knowledge about each such domain. As a result, FERMI's knowledge and problem-solving power can be greatly extended, with a minimum of effort, to yield centrally important principles and associated methods for dealing with problems in a variety of domains. The following paragraphs summarize briefly some of the many important domain-specific principles and problems that FERMI can potentially subsume by its general knowledge about decomposability and invariance.

5.2.1. Domain principles subsumed by decomposability

The general principle of decomposability is applicable to many more quantities and domains than indicated by the examples in the preceding sections. Most often the decomposability of a quantity is achieved by simple additivity (although it may be more complex, as already illustrated in our example of center of mass). The following list mentions some of the domains in which the decomposability principle is applicable to some specific quantity (indicated by *italics*). In each case, the decomposability principle subsumes an important domain-specific principle asserting that the particular quantity is decomposable; correspondingly, this principle then allows the solution of an important class of problems in this domain.

- *Hydrostatics* (liquids at rest). The *pressure drop* between any two points in a liquid can be decomposed into a sum of pressure drops along successive segments of any path joining the two points. (This principle, and correspondingly solvable problems, have already been discussed.)

- *Electric circuits.* The *potential drop* between any two points in a circuit can be decomposed into a sum of potential drops along successive circuit paths joining the points. (This principle, and correspondingly solvable problems, have also already been discussed in the preceding sections.)

- *Geometry.* The *length* of a curved line can be decomposed into the sum of lengths of successive segments of this line. Similarly, the *area* of a surface can be decomposed into a sum of areas of component elements of this surface. Similarly, the *volume* of a region can be decomposed into the sum of volumes of component elements of this region. All these principles are, of course, very familiar and widely used in geometry

5. Potential Capabilities

FERMI's capabilities may be substantially enhanced by relatively small additions to it's knowledge base. We now describe some of the possible and planned extensions of FERMI's knowledge and correspondingly increased capabilities. We discuss these extensions in order of increasing scope. We first describe extensions involving additional domain-specific knowledge subsumed by the present general *principles of decomposability and invariance, then extensions involving additional* general principles, and then extensions involving more general methods. Finally, we mention extensions designed to provide FERMI with greater explanatory power and with teaching capabilities.

5.1. Present Knowledge and Capabilities

The current prototype FERMI system implements only very limited knowledge. In particular, FERMI's general knowledge includes only two general principles (decomposability and invariance) and associated general methods (decomposition and comparison of invariants), together with a knowledge of algebra. Nevertheless, this limited knowledge, when used in conjunction with a small amount of domain-specific knowledge about three different domains, is sufficient to enable FERMI to solve a fairly large range of reasonably complex problems. As discussed in the preceding sections, the types of problems, presently solvable by FERMI, include the following:

- *Problems about pressures in liquids.* FERMI can find the pressure difference between any two points in one or more liquids at rest, even when there are several distinct layers of such liquids, and even if these liquids are in a container whose walls interrupt a direct straight line connecting the points of interest.

- *Problems about centers of mass.* FERMI can find the center of mass of any planar object which is rectangular or decomposable into rectangular parts.

- *Problems about electric circuits.* FERMI can find potential drops or (time-independent) currents in electric circuits consisting of any small number of wires, resistors, and batteries interconnected in various ways.

5.2. Applications of Present Knowledge to Additional Domains

FERMI's present knowledge includes two very general principles, decomposability and invariance, specifying commonly occurring and important properties of certain quantities. These general principles subsume more domain-specific principles (e.g., principles about pressure drops in liquids or potential drops in electric circuits) that are

invariance, FERMI generates two expressions equal to potential-drop-1: (1) Expression-2 is potential-drop-3 from *b* to *c* through the battery. (2) *Expression-3* is potential-drop-7 from *b* to *c* along the path through resistor R_2. Third (G13 R13), FERMI evaluates one of these expressions to yield a value for potential-drop-1. Details of this work are given in panels 3 and 4.

In panel 3 of Figure 11, FERMI evaluates expression-3, the first of its expressions for potential-drop-1. This evaluation succeeds (R14) and so satisfies goal G13 to evaluate *some* expression for potential-drop-1.

Because expression-3 equals potential-drop-3 (from *b* to *c* through the battery) goal G14 to evaluate expression-3 requires only goal G15 to find potential-drop-3. G15 is achieved in the following three steps: First, pullers fail (G16-R16) because the path from b' to c' is not a single component, but composed of two wires plus a battery. Second, applying methods (G17-R17), known path decomposition (kpd) succeeds. The reason is that when FERMI constructs paths through circuits, it constructs them in terms of components. Therefore, when constructing the alternate path from *b* to *c* through the battery, FERMI also constructed components of this path. Constructing these components is FERMI's way of distinguishing between alternative paths with the same endpoints. It also now allows FERMI to proceed immediately to decompose this path into already-known components. Third, (G18-R18) FERMI evaluates expression(s) produced by the methods. This evaluation succeeds, with details given in Panel 4.

Panel 4 evaluates expression-5 produced by the kpd method applied to the path from *b* to *c* through the battery. This expression-5 is the sum of the three potential drops (2 wires, 1 battery) along the path from *b* to *c* through the battery (see Figure 12). Therefore the work in Panel 4 divides into the following three parts: (G20-R20), finding potential-drop-6 from *b* to *a*, (G22-R22) finding potential-drop-5 from *a* to *d* across the battery, and (G27-R27) finding potential-drop-4 from *d* to *c*. The first and third of these steps are easily handled by a puller that returns the value 0 for a wire. The second step (G22-R22) is also handled by a puller in two steps: (G23-R23) finding an expression (expression-6: emf-1) for the potential drop, and (G24-R24) evaluating that expression by looking up the value of emf-1 (the emf of the battery).

When the values (0.0, 6.0, 0.0) of these three potential drops have been found, they are combined by addition (as indicated in G19) to yield the value (6.0) of expression-5.

node *b*. (The other node through which I_0 flows is node *a* involving just one other current, that flowing through the battery. FERMI prefers to apply input-output invariance to nodes with larger numbers of inputs and outputs.) Applying input-output invariance to node *b* produces *expression-1* (+ current-1 current-2), where current-1 and current-2 are respectively the currents flowing through resistors R_1 and R_2. In (G4-R4) FERMI evaluates the single expression generated by relevant methods. This evaluation is successful, yielding the desired value of current-0.

In more detail, the goal G4 to evaluate any available expression for current-0 sets the more specific subgoal (G5) to evaluate expression-1 (+ current-1 current-2). Goal G5 requires the *AND* subgoal to find values for *both* current-1 and current-2. These values for are found by identical processes given in the detailed traces in panel 2 and subsequent panels.

We now turn to panel 2 to elaborate FERMI's process for getting a value for current-1, i.e. for getting result R6 in response to goal G6. There are two main steps in this part of the trace: (1) Applying pullers (G7-R7) yields *expression-2* = (*I* potential-drop-1 resistance-1) where potential-drop-1 is the potential drop from *b* to *c* across resistor 1 and resistance-1, is the resistance of that resistor. (2) Evaluating this expression (G8-R8) yields the value of current-1.

In more detail, the computability requirements of the 'current' puller are satisfied because I_1 is a current through a single component (the resistor labeled R_1 in Figure 12). FERMI uses this puller to produce expression-2. The goal G8 is the goal to evaluate all available expressions for I_1, and G9 aims to evaluate expression-2. Evaluating expression-2 requires the *AND* goal to find values for both potential-drop-1 and resistance-1. These values are found through a simple lookup for resistance-1 (G29-R29) and through application of methods for potential-drop-1 (G10-R10).

The goal G10 to find potential-drop-1 is achieved in three major steps: First with (G11-R11), FERMI tries to apply the puller potential-drop = (* resistance current). This relation, however, is the same as that just applied to I_1 under goal G7. Applying it again to find potential-drop-1 in terms of I_1 simply generates a new copy of the old equation. FERMI therefore rejects application of this puller with the message *non-circularity: not satisfied*. This message means that the puller expression contains one or more quantities (here I_1) already encountered in the tree of goals. Applying this puller a second time would produce algebraic circularity. Second (G12-R12), FERMI applies methods which produce expression-3 and expression-4 for potential-drop-1. All of the decomposition methods (kpd, ipd, rpd) fail because the path considered is a single circuit element, and cannot be decomposed. Using path

Figure 12:   Circuit with two resistors in parallel.
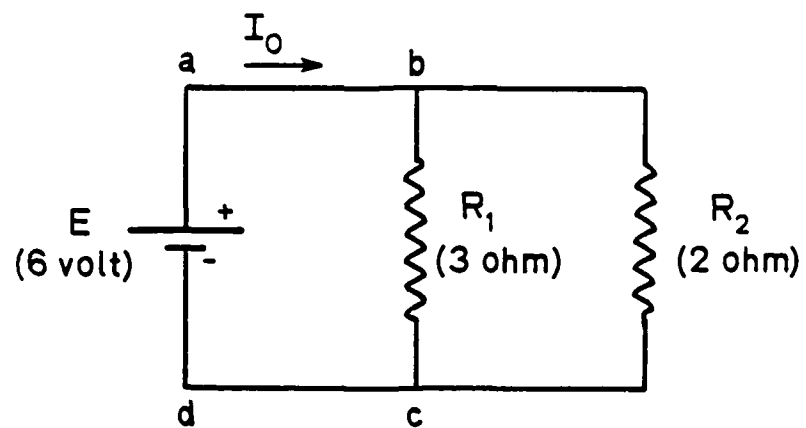
Panel 4
Evaluating the Expression for the Sum of Potential Drops

(Figure 11 continued)

```
G19:  Evaluate expression-5: (+ potential-drop-6
      |   potential-drop-5 potential-drop-4)
      |  AND { potential-drop-6 potential-drop-5 potential-drop-4 }
      |   G20: potential-drop-6
      |   |  Lookup: empty
      |   |  G21: Apply-pullers
      |   |  |  Apply puller: (* current resistance) or (- emf))
      |   |  |  | Computability-requirements: satisfied
      |   |  |  potential-drop-6 = 0
      |   |  R21: potential-drop-6 = 0
      |   R20: potential-drop-6 = 0
      |   G22: potential-drop-5
      |   |  Lookup: empty
      |   |  G23: Apply-pullers.
      |   |  Apply puller:
      |   |  (* current resistance) or (- emf))
      |   |   Computability-requirements: satisfied
      |   |  potential-drop-5 = expression-6 [emf-1]
      |   |  R23: potential-drop-5 = (expression-6)
      |   |  G24: evaluate-expressions-for-potential-drop-5
      |   |  |   OR {   expression-6 }
      |   |  |    G25: Evaluate expression-6: (emf-1)
      |   |  |    |   AND { emf-1 }
      |   |  |    |   G26: emf-1
      |   |  |    |   |Lookup: 6.0
      |   |  |    |   R26: emf-1 = 6.0
      |   |  |    |  expression-6 = 6.0
      |   |  |    R25: expression-6 = 6.0
      |   |  R24: evaluate-expressions-for-potential-drop-5:   6.0
      |   R22: potential-drop-5 = 6.0
      |   G27: potential-drop-4
      |   |  Lookup: empty
      |   |  G28: Apply-pullers
      |   |  Apply puller:
      |   |   (* current resistance) or (- emf)
      |   |    Computability-requirements:  satisfied
      |   R28: potential-drop-4 = 0
      |   R27: potential-drop-4 = 0
      R19: expression-5 = 6.0
```

Panel 3
Evaluating the Potential Drop from *b* to *c*

(Figure 11 continued)

G13: evaluate-expressions-for-potential-drop-1
  OR { expression-3 expression-4 }
  G14: Evaluate expression-3: (potential-drop-3)
        AND { potential-drop-3 }
        G15: potential-drop-3
        Lookup: empty
        G16: Apply-pullers
        Apply puller:
         (* current resistance) or (- emf))
         path-lies-in-same-component not-satisfied
        fails
        R16: fails
        G17: Apply methods   { kpd ipd rpd path-invariance }
        Apply method: kpd
        potential-drop-3 = expression-5
                [+ potential-drop-6   {from *b* to *a*}
                    potential-drop-5   {from *a* to *d*}
                    potential-drop-4]   {from *d* to *c*}
        R17: potential-drop-3 = expression-5
        G18: evaluate-expressions-for-potential-drop-3
            OR {   expression-5 }
            G19: Evaluate expression-5:
                 (+ potential-drop-6
                    potential-drop-5 potential-drop-4)

                            •
                            •
                            •

                    Elaborated in panel 4
            R19: expression-5 = 6.0
        R18: evaluate-expressions-for-potential-drop-3: 6.0
        R15: potential-drop-3 = 6.0
  R14: expression-3 = 6.0
R13: evaluate-expressions-for-potential-drop-1:   6.0

Panel 2:   Finding current-1

(Figure 11 continued)

G6: current-1
| Lookup: empty
| **G7: Apply-pullers**
| | Apply puller: (/ potential-drop resistance)
| | | Computability-requirements: satisfied
| | current-1 = expression-2 [/ potential-drop-1 resistance-1]
| **R7: current-1 = (expression-2)**
| **G8: evaluate-expressions-for-current-1**
| | | OR {   expression-2 }
| | | G9: Evaluate expression-2: (/ potential-drop-1 resistance-1)
| | | | AND { potential-drop-1 resistance-1 }
| | | | G10: potential-drop-1
| | | | | Lookup: empty
| | | | | G11: Apply-pullers
| | | | | | Apply puller: (* current resistance) or (- emf))
| | | | | | | visited-quantities not-satisfied
| | | | | | fails
| | | | | R11: fails
| | | | | G12: Apply methods { kpd ipd rpd path-invariance }
| | | | | | Apply method: kpd
| | | | | | Fails
| | | | | | Apply method: ipd
| | | | | | Fails
| | | | | | Apply method: rpd
| | | | | | Fails
| | | | | | Apply method: path-invariance
| | | | | | potential-drop-1 = expression-3 [potential-drop-3]
| | | | | | {from *b* to *c* through battery}
| | | | | | = expression-4 [potential-drop-7]
| | | | | | {from *b* to *c* through resistor R_2}
| | | | | R12: potential-drop-1 = expression-3
| | | | | | = expression-4
| | | | | G13: evaluate-expressions-for-potential-drop-1
| | | | | | :
| | | | | | *Elaborated in panel 3*
| | | | | R13: evaluate-expressions-for-potential-drop-1:   6.0
| | | | R10: potential-drop-1 = 6.0
| | | | G29: resistance-1
| | | | | Lookup: 3.0
| | | | R29: resistance-1 = 3.0
| | | R9: expression-2 = 2.0
| **R8: evaluate-expressions-for-current-1:   2.0**
R6: current-1 = 2.0

Figure 11:   Trace of FERMI's work on a complex problem
requiring several kinds of knowledge.

Problem: Two resistors of resistance R_1 = 3 ohm and R_2 = 2 ohm are
connected in parallel as shown in Figure 9.   The battery has an emf of 6 volts.
What is the current I_0 flowing through the wire from a to b?

Panel 1
Main Steps of the Solution

G1: current-0
Lookup: empty
G2: Apply-pullers
Apply puller: (/ potential-drop resistance)
| finite-resistance not-satisfied
fails
R2: fails
G3: Apply methods
Apply method: input-output-invariance
current-0 = expression-1 [+ current-1 current-2]
R3: current-0 = expression-1
G4: evaluate-expressions-for-current-0
OR {   expression-1 }
G5: Evaluate expression-1: (+ current-1 current-2)
AND { current-1 current-2 }
G6: current-1
•
•
•
This part of trace elaborated in panel 2.
R6: current-1 = 2.0
G7: current-2
•
•
•
Finding current-2 proceeds exactly like finding current-1.
R7: current-2 = 3.0
R5: expression-1 = 5.0
R4: evaluate-expressions-for-current-0:   5.0
R1: current-0 = 5.0

structure would become appreciably simpler since many of the currently encountered complexities would be shifted to algebraic manipulations. Correspondingly, FERMI should thereby become able to solve more complex problems with relative simplicity.

5.4. Problems Solvable with Minor Extensions

Even with minor extensions of its present knowledge base, FERMI would be able to solve a significant number of important problems in a variety of domains. The following are examples.

- *Electric circuits.* FERMI already incorporates all the essential knowledge needed to deal with direct-current electric circuits. This knowledge includes the decomposability and invariance principle applied to potential drops along any circuit path, the invariance principle applied to currents flowing into or out of any circuit junction point, and domain-specific knowledge about the potential drop across a resistor or battery. With improved methods of constraint satisfaction, FERMI should then be able to solve electric-circuit problems of any complexity. In other words, it should be able to find the current flowing in any circuit element, or the potential drop across any circuit element, in any interconnected set of batteries and resistors. With the same knowledge FERMI could also find the internal resistance of a non-idealized battery from appropriate currents and potential drops in the circuit.

- *Hydraulic systems.* A hydraulic system, consisting a liquid flowing in an interconnected set of pipes and pumps, is analogous to an electric circuit (with pipes being analogous to resistors, and pumps analogous to batteries). With appropriate translation of its electric-circuit knowledge to this new domain, FERMI should thus be able to solve all kinds of problems involving hydraulic systems.

- *Hydrostatics.* FERMI already has the capability of finding the pressure drop between any two points in a liquid, or set of liquids, at rest. With the addition of simple domain-specific knowledge relating pressure to force and area, FERMI can then easily find forces on various surfaces. Suppose that FERMI is also supplied with the general knowledge that its additive decomposability principle is applicable to forces, and that the total force on any object in equilibrium satisfies the constraint $F = 0$. Then FERMI should be able to solve quite complex problems involving liquids at rest, including problems involving objects floating in one or more liquids or floating on liquid surfaces.

- *Chemistry.* With relatively small additions of domain-specific knowledge, FERMI should be able to cope with various problems in chemistry. For example, problems in "stoichiometry" (the relative proportions of substances involved in chemical reactions) should be amenable to FERMI

by using the additive decomposability principle relating the mass of a molecule to the masses of its constituent atoms, and by using the invariance principle applied to the number of atoms of each kind involved in a chemical reaction.   Furthermore, as mentioned previously, simple problems about reaction rates should also become accessible by applying the decomposability principle to relate a reaction rate to the concentrations of individual reagents.

5.5. Additions to FERMI's General Methods

5.5.1. Semantic interpretation of algebra

FERMI already has substantial algebraic capabilities.   By applying general statements about principles to information about a specific problem, it can translate the results into the form of equations.   However, it would be desirable to have the converse ability to interpret the meaning of equations obtained during a solution process.   This would allow FERMI to use algebra more intelligently and efficiently, and also to explain better any results obtained by its calculations.   FERMI would then also simulate more closely the behavior of human experts whose problem-solving abilities are often enhanced by the ability to interpret collections of algebraic symbols (Larkin & Simon, 1981).

For example, consider the following problem illustrated in Figure 14:

A one-meter wide mine shaft is slanted so that its top is horizontally offset by 2 meters from its bottom 4 meters below.   This shaft is filled to its top with water having a density ρ.   What is the pressure drop from a point A, at the center of the bottom of the shaft, to a point B at the center of the top of this shaft?

FERMI solves this problem by joining the points A and B by the path indicated by the dashed lines in Figure 14, calculating the pressure drops along each of the six vertical or horizontal path segments, and then adding the results. However, a simpler and more transparent solution is produced by the following use of algebra with appropriate interpretation.   Direct application of FERMI's principles leads to the following result for the desired pressure drop P:

$$P = \Sigma_i \, \rho \, g \, h_i,$$

where h_i is the vertical height of each path segment.   But this expression can be simplified in the following way:

Figure 14:   Mine-shaft problem facilitated by the interpretation of equations.
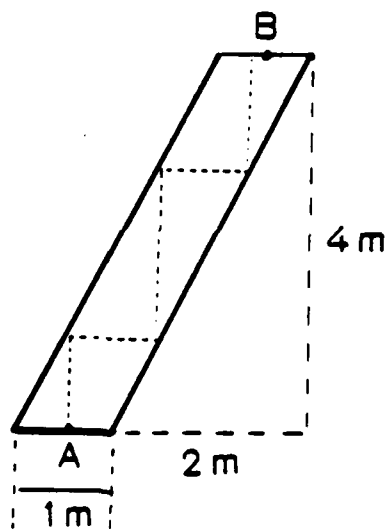
$$P = \rho g \sum_i h_i = \rho g \sum_{i \ (vertical)} h_i$$
$$= \rho g H.$$

The successive steps in this simplification involve the following: (1) Taking the common factors ρ and g outside the summation. (2) Omitting all terms for the horizontal segments since the height h_i is zero for each. (3) Replacing the remaining sum over individual heights by H, which can be interpreted as the total vertical height separating the points A and B.

One algebraic interpretation ability displayed in this solution involves the omission of terms that are equal to zero. The more sophisticated and important interpretation ability is that demonstrated in the third step. To implement this step, FERMI would need to use its present knowledge that height is a decomposable quantity, obtainable by addition of component heights, in order to infer that the sum of vertical heights, appearing in an equation, can be interpreted as the total height separating two points. Such an interpretive ability would not only simplify the solution of this problem. It would also lead FERMI to display the intelligent insight that the answer to this problem depends merely on the vertical separation of the points, but does *not* depend on the width of the shaft or on the horizontal offset between its top and bottom.

5.5.2. Reasoning by analogy.

FERMI's abilities would be improved by the ability to recognize and exploit analogies, both within a given problem and between different problems.

The potential utility of analogical reasoning *within* a given problem is illustrated by the following example. When FERMI is currently faced with a circuit problem involving two identical branches, (as in Figure 12), it carries out a calculation to find the current in the first branch and then carries out exactly the same calculation again for the second branch. The ability to recognize that the two branches are analogous would allow FERMI simply to take its solution for the first branch and map it onto the other, without the need to repeat the calculation.

The potential utility of analogical reasoning *between* different problems is illustrated by the following example. When FERMI is currently faced by a problem asking for the pressure drop between two points which are not vertically aligned, it joins these points by a path consisting of vertical and horizontal segments. It then sums potential drops for each segment, simply adding zero for each horizontal segment, so that only vertical segments are relevant. But FERMI does not make this inference explicitly, nor is it able to use this result to solve more simply other problems involving points that are not vertically aligned. Improved design should allow

FERMI to store its knowledge about a previously solved problem so that it can then map corresponding results and methods to other problems recognized as analogous.

5.6. Applications to Understanding and Learning by Humans

FERMI offers the promise of improving human-computer interaction and of providing some insights into effective forms of human knowledge organization. Such applications include the following:

- *Effective explanatory power.* FERMI's hierarchical knowledge organization, separating general knowledge from more detailed domain-specific knowledge, should help FERMI to explain its reasoning in ways that are easily understandable by humans. FERMI can explain its reasoning by referring to a few general principles that are likely to be familiar to humans. Many of the general principles (invariance, decomposition) have analogs in everyday life and may be somewhat familiar from a variety of contexts. Furthermore, such general knowledge can serve as an embedding framework and advance organizer for explanations involving more detailed domain-specific knowledge.

- *Instructional applications.* FERMI should be useful for the design of effective computer-aided instruction or intelligent tutoring systems. In particular, FERMI has the advantage that it can present its knowledge and explain its reasoning in forms comprehensible to humans. Furthermore, FERMI can induce a human learner to acquire knowledge organized in useful forms separated according to different levels of generality.

- *Psychological studies of human knowledge organization.* FERMI can provide a computer-implemented psychological model for studying the effects of human knowledge organizations that structure knowledge hierarchically and separate knowledge of different degrees of generality. Furthermore, irrespective of any computer involvement, it is possible to carry out experiments on human subjects who have acquired similar knowledge organizations. The effectiveness of such human knowledge organizations can then be studied by investigating the performance of such subjects on various intellectual tasks.

Applications to explanation and human learning depend strongly on FERMI's mechanisms for storing problem solutions and producing explanatory traces of its work. The traces given in this paper (Figures 8, 9, and 12 were produced from a hierarchical tree structure reflecting the goal hierarchy of the problem solution.) Figure 15 shows the main entries in these trees. These trees are formed as FERMI solves a problem, and then used to construct traces. Because the trees are hierarchical, FERMI has the potential to produce solution descriptions at varying levels of detail,

depending on what it is communicating and to whom. These hierarchical traces may also form the basis for new abilities to generate analogies between problems so as to apply some of the more general features of one problem solution to a new situation.
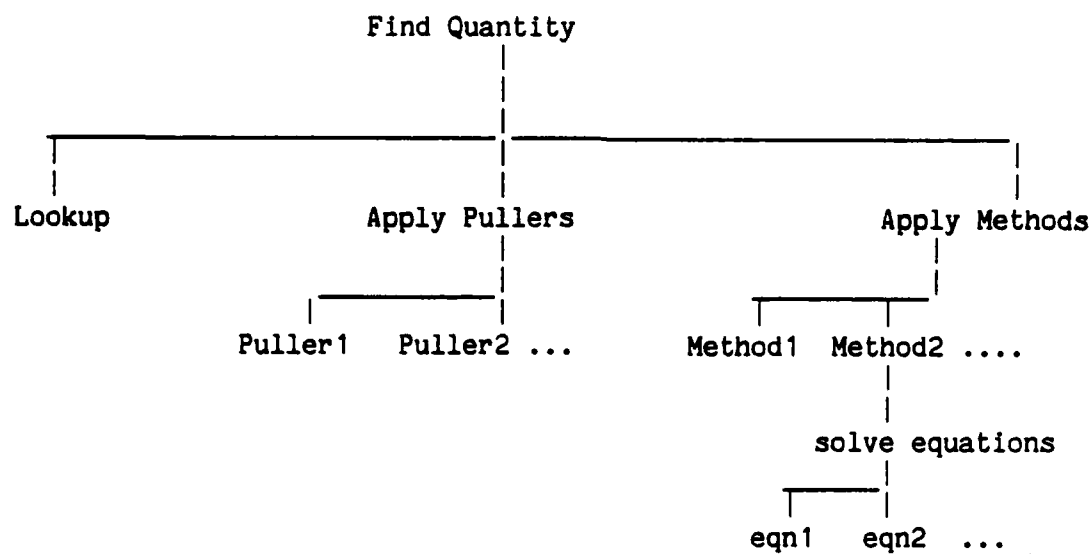
6. Concluding Remarks

The preceding pages discuss the design of an expert system which separates its knowledge according to levels of generality, embedding domain-specific knowledge within more general knowledge. Such a knowledge representation is economical and powerful, having the following advantages: (1) General knowledge can be applied repeatedly in different domains, with only limited additions of more domain-specific knowledge. (2) Knowledge can be used flexibly, with relatively easy generalization or transfer to new domains. Correspondingly, rather small amounts of knowledge can interact in rich ways to solve a large number of varied problems. (3) Such a system can exploit its general knowledge to communicate more readily with humans in order to explain its reasoning or to teach.

FERMI is a prototype expert system based on these design ideas. In particular, FERMI's knowledge is organized into schemas, hierarchically organized according to levels of generality, with inheritance of information from more general schemas, higher in the hierarchy, to more specific schemas at lower levels. There are two such interacting hierarchies: One of these encompasses declarative knowledge about various entities, including general principles specifying the properties of important quantities; the other encompasses associated procedural knowledge, including general methods to be used in conjunction with important principles.

The general knowledge, included in the present prototype implementation of FERMI, is quite limited. It includes essentially only two general principles (decomposability and invariance), together with associated general methods (decomposition and comparison of invariants) and algebraic capabilities. Nevertheless, when combined with relatively small amounts of domain-specific information, this limited general knowledge allows FERMI to solve a considerable range of fairly complex problems in several different domains (e.g., finding pressure drops in liquids, potential drops or currents in electric circuits, or centers of mass of complexly-shaped objects).

The present prototype implementation of FERMI should be readily extensible by adding a few more general principles and methods. When these are used jointly in conjunction with limited amounts of domain- specific knowledge, FERMI promises to become a powerful expert system capable of solving a far larger range of problems in various domains. It should then also be readily able to explain its reasoning to

Figure 15:   Structure of FERMI's traces.


```
                        Find Quantity
                              |
                              |
                              |
        |_____|_____|
        |                     |                           |
     Lookup             Apply Pullers               Apply Methods
                              |                            |
                     |_____|                    |_____|
                     |        |                    |       |
                  Puller1  Puller2 ...         Method1  Method2 ....
                                                            |
                                                            |
                                                    solve equations
                                                     |_____|
                                                     |      |
                                                   eqn1    eqn2   ...
```

humans and should provide a useful basis for an effective intelligent tutoring system.

References

Bobrow, D. G. and Winograd, T. (1977). An Overview of KRL, a Knowledge Representation Language. *Cognitive Science, 1*(1), 3-46.

Brachman, R. J. (1979). On the Epistemological Status of Semantic Networks. In N. V. Findler (Ed.), *Associative Networks*. New York: Academic Press.

Brown, A.L. & Campione J.C. (1984). Three Faces of Transfer: Implications for Early Competence, Individual Differences, and Instruction. In M. Lamb, A. Brown & B. Rogoff (Eds.), *Advances in Developmental Psychology*. Hillsdale, N.J.: Laurence Erlbaum Associates.

Chi, M.T.H., Feltovich, P.J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5*(2), 121-152.

Fahlman, S. E. (1979). *NETL: A System for Representing and Using Real World Knowledge*. MIT Press.

Feigenbaum, E. A., Buchanan, E. A. and Lederberg, J. (1971). On Generality and Problem Solving: A Case Study Using the DENDRAL Program. In D. Michie (Ed.), *Machine Intelligence 6*. Edinburgh University Press.

Fox, M. S. (1979). On Inheritance in Knowledge Representation. In *IJCAI VI Proceedings*. IJCAI-79.

Larkin, J.H. (1981). Enriching formal knowledge: A model for learning to solve problems in physics. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Larkin, J.H. (1983). The role of problem representation in physics. In D. Gentner & A.L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Larkin, J.H. & Simon, H.A. (August 19-21 1981). Learning through growth of skill in mental modeling. In *Proceedings of the Third Annual Conference*. Berkeley, CA: Cognitive Science Society.

Larkin, J.H., McDermott, J., Simon, D.P., & Simon, H.A. (June 1980). Expert and novice performance in solving physics problems. *Science, 208*, 1335-1342.

Larkin, J.H., McDermott, J., Simon, D.P., & Simon, H.A. (1980). Models of competence in solving physics problems. *Cognitive Science, 4*, 317-345.

McDermott, J. (1980). *R1: A Rule-Based Configurer of Computer Systems* (Tech. Rep.). Carnegie-Mellon University, Computer Science Department.

McDermott, J. (1982). XSEL: A Computer Salesperson's Assistant. In Hayes, J., Michie, D. and Pao, Y-H. (Eds.), *Machine Intelligence 10*. Chichester UK: Ellis Horwood Ltd.

McDermott, J. & Larkin, J. H. (1978). Representing textbook physics problems. In *Proceedings of the 2nd national conference*. The Canadian society for computational studies of intelligence.

Minsky, M. (1975). A Framework for Representing Knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.

Reed, S. K., Ernst, G.W. & Banerji, R. (1974). The role of analogy in transfer between similar problem states. *Cognitive Psychology, 6*, 436-450.

Reif, F. & Heller, J.I. (1982). Knowledge structure and problem solving in physics. *Educational Psychologist, 17*, 102-127.

Shortliffe, E. (1976). *Computer Based Medical Consultations: MYCIN*. New York: Elsevier.

Simon, D.P. & Simon, H.A. (1978). Individual differences in solving physics problems. In R. Siegler (Ed.), *Children's thinking: What develops?*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sussman, G. J. (1975). *A Computer Model of Skill Acquisition*. New York: Elsevier. (MIT PhD thesis in 1973).

Wright, K. and Fox, M. (1983). *The SRL Users Manual* (Tech. Rep.). Robotics Institute, Carnegie-Mellon University.

Distribution list

A Flexible Expert Reasoner with Multi-Domain Inferencing

Personnel Analysis Division
AF/MPXA
5C360. The Pentagon
Washington. DC 20330

Air Force Human Resources Lab
AFHRL/MPD
Brooks AFB. TX 78235

Air Force Office
   of Scientific Research
Life Sciences Directora
Bolling Air Force
Washington. DC 20332

Dr. Robert Ahlers
Code N711
Human Factors Laboratory
NAVTRAEQUIPCEN
Orlando. FL 32813

Dr. Ed Aiken
Navy Personnel R&D Center
San Diego. CA 92152

Dr. William E. Alley
AFHRL/MOT
Brooks AFB. TX 78235

Dr. Earl A. Alluisi
HQ. AFHRL (AFSC)
Brooks AFB. TX 78235

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh. PA 15213

Dr. John Annett
University of Warwick
Department of Psychology
Coventry CV4 7AJ
ENGLAND

Special Assistant for Project
OASN(M&RA)
5D800. The Pentagon
Washington, DC 20350

Dr. Michael Atwood
ITT - Programming
1000 Oronoque Lane
Stratford. CT 06497

Dr. Alan Baddeley
Medical Research Council
Applied Psychology Unit
15 Chaucer Road
Cambridge CB2 2EF
ENGLAND

Dr. Patricia Baggett
University of Colorado
Department of Psychology
Box 345
Boulder, CO 80309

Dr. Eva L. Baker. Director
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles. CA 90024

Mr. J. Barber
HQS. Department of the Army
DAPE-ZBR
Washington. DC 20310

Capt. J. Jean Belanger
Training Development Division
Canadian Forces Training System
CFTSHQ. CFB Trenton
Astra. Ontario. KOK
CANADA

/

Technical Director
Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria. VA 22333

Dr. Werner Birke
Personalstammamt
  der Bundeswehr
D-5000 Koeln 90
WEST GERMANY

Dr. Gautam Biswas
Department of Computer Science
University of South Carolina
Columbia. SC 29208

Dr. Alvah Bittner
Naval Biodynamics Laboratory
New Orleans. LA 70189

Dr. John Black
Yale University
Box 11A. Yale Station
New Haven. CT 06520

Code N711
Attn: Arthur S. Blaiwes
Naval Training Equipment Center
Orlando. FL 32813

Dr. Robert Blanchard
Navy Personnel R&D Center
San Diego. CA 92152

Mr. Arnold Bohrer
Psychological Research Section
Caserne Petits Chateau
CRS
1000 Brussels
BELGIUM

Dr. Jeff Bonar
Learning R&D Center
University of Pittsburgh
Pittsburgh. PA 15260

CDR Robert J. Biersner. USN
Naval Biodynamics Laboratory
P. O. Box 29407
New Orleans. LA 70189

Dr. Menucha Birenbaum
School of Education
Tel Aviv University
Tel Aviv. Ramat Aviv 69978
Israel

Dr. Gordon H. Bower
Department of Psychology
Stanford University
Stanford. CA 94306

Dr. Richard Braby
NTEC Code 10
Orlando. FL 32751

Dr. Robert Breaux
Code N-095R
NAVTRAEQUIPCEN
Orlando. FL 32813

Dr. John S. Brown
XEROX Palo Alto Research
Center
3333 Coyote Road
Palo Alto. CA 94304

Dr. Bruce Buchanan
Computer Science Department
Stanford University
Stanford. CA 94305

Dr. Patricia A. Butler
NIE Mail Stop 1806
1200 19th St.. NW
Washington. DC 20208

Dr. Robert Calfee
School of Education
Stanford University
Stanford. CA 94305

Col. Roger Campbell
AF/MPXOA
Pentagon, Room 4E195
Washington, DC 20330

Dr. Nick Bond
Office of Naval Research
Liaison Office. Far East
APO San Francisco. CA 96503

Dr. Lyle Bourne
Department of Psychology
University of Colorado
Boulder. CO 80309

Dr. Susan Carey
Harvard Graduate School of
    Education
337 Gutman Library
Appian Way
Cambridge. MA 02138

Dr. Pat Carpenter
Carnegie-Mellon University
Department of Psychology
Pittsburgh. PA 15213

Dr. Robert Carroll
NAVOP 01B7
Washington. DC 20370

Dr. Fred Chang
Navy Personnel R&D Center
Code 51
San Diego. CA 92152

Dr. Davida Charney
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh. PA 15213

Dr. Eugene Charniak
Brown University
Computer Science Department
Providence. RI 02912

Dr. Michelene Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh. PA 15213

Dr. Jaime Carbonell
Carnegie-Mellon University
Department of Psychology
Pittsburgh. PA 15213

Mr. James W. Carey
Commandant (G-PTE)
U.S. Coast Guard
2100 Second Street. S.W
Washington. DC 20593

Director
Manpower Support and
    Readiness Program
Center for Naval Analysis
2000 North Beauregard Street
Alexandria. VA 22311

Scientific Advisor
    to the DCNO (MPT)
Center for Naval Analysis
2000 North Beauregard Street
Alexandria. VA 22311

Chief of Naval Education
    and Training
Liaison Office
Air Force Human Resource Laboratory
Operations Training Division
Williams AFB. AZ 85224

Assistant Chief of Staff
Research. Development.
Test. and Evaluation
Naval Education and
Training Command (N-5)
NAS Pensacola. FL 32508

Dr. Michael Cole
University of California
    at San Diego
Laboratory of Comparative
    Human Cognition - D003A
La Jolla. CA 92093

Dr. Allan M. Collins
Bolt Beranek & Newman. Inc.
50 Moulton Street
Cambridge. MA 02138

Dr. Susan Chipman
Code 442PT
Office of Naval Research
800 N. Quincy St.
Arlington. VA 22217-5000

Dr. Stanley Collyer
Office of Naval Technology
800 N. Quincy Street
Arlington. VA 22217

Mr. Raymond E. Christal
AFHRL/MOE
Brooks AFB. TX 78235

Dr. Lynn A. Cooper
Learning R&D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh. PA 15213

Dr. William Clancey
Computer Science Department
Stanford University
Stanford. CA 94306

Dr Meredith P Crawford
American Psychological Association
Office of Educational Affairs
1200 17th Street. N.W.
Washington. DC 20036

Mr. Paul DiRenzo
Commandant of the Marine Corps
Code LBC-4
Washington. DC 20380

Dr. R. K. Dismukes
Associate Director for Life Sciences
AFOSR
Bolling AFB
Washington. DC 20032-6448

Dr. Hans Crombag
University of Leyden
Education Research Center
Boerhaavelaan 2
2334 EN Leyden
The NETHERLANDS

Dr. Emmanuel Donchin
University of Illinois
Department of Psychology
Champaign. IL 61820

Dr. Lee Cronbach
16 Laburnum Road
Atherton. CA 94205

Dr. Kenneth B. Cross
Anacapa Sciences. Inc.
P.O. Drawer Q
Santa Barbara. CA 93102

Defense Technical
Information Center
Cameron Station. Bldg 5
Alexandria. VA 22314
Attn: TC
(12 Copies)

Dr. Mary Cross
Department of Education
Adult Literacy Initiative
Room 4145
400 Maryland Avenue. SW
Washington. DC 20202

Dr. Thomas M. Duffy
Communications Design Center
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

CTB/McGraw-Hill Library
2500 Garden Road
Monterey. CA 93940

Barbara Eason
Military Educator's
Resource Network
InterAmerica Research Associates
1555 Wilson Blvd.
Arlington. VA 22209

CDR Mike Curran
Office of Naval Research
800 N. Quincy St.
Code 270
Arlington, VA 22217-5000

Bryan Dallman
AFHRL/LRT
Lowry AFB, CO 80230

Dr. Charles E. Davis
Personnel and Training Research
Office of Naval Research
Code 442PT
800 North Quincy Street
Arlington, VA 22217-5000

Mr. Robert Denton
AFMPC/MPCYPR
Randolph AFB, TX 78150

Dr. Richard Elster
Deputy Assistant Secretary
  of the Navy (Manpower)
Washington, DC 20350

Dr. Susan Embretson
University of Kansas
Psychology Department
Lawrence, KS 66045

Dr. Randy Engle
Department of Psychology
University of South Carolina
Columbia, SC 29208

Lt. Col Rich Entlich
HQ, Department of the Army
OCSA(DACS-DPM)
Washington, DC 20310

Dr. K. Anders Ericsson
University of Colorado
Department of Psychology
Boulder, CO 80309

Edward Esty
Department of Education, OERI
MS 40
1200 19th St., NW
Washington, DC 20208

Edward E. Eddowes
CNATRA N301
Naval Air Station
Corpus Christi, TX 78419

Dr. John Ellis
Navy Personnel R&D Center
San Diego, CA 92252

Dr. Jeffrey Elman
University of California,
San Diego
Department of Linguistics, C-008
La Jolla, CA 92093

Mr. Wallace Feurzeig
Educational Technology
Bolt Beranek & Newman
10 Moulton St.
Cambridge, MA 02238

Dr. Craig I. Fields
ARPA
1400 Wilson Blvd.
Arlington, VA 22209

Dr. Gerhard Fischer
Liebiggasse 5/3
A 1010 Vienna
AUSTRIA

Dr. Dexter Fletcher
University of Oregon
Computer Science Department
Eugene, OR 97403

Dr. Linda Flower
Carnegie-Mellon University
Department of English
Pittsburgh, PA 15213

Dr. Ken Forbus
Department of Computer Science
University of Illinois
Champaign, IL 61820

Dr. Beatrice J. Farr
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Marshall J. Farr
2520 North Vernon Street
Arlington, VA 22207

Dr. Pat Federico
Code 511
NPRDC
San Diego, CA 92152

Dr. Paul Feltovich
Southern Illinois University
School of Medicine
Medical Education Department
P.O. Box 3926
Springfield, IL 62708

Dr. Alfred R. Fregly
AFOSR/NL
Bolling AFB, DC 20332

Dr. Bob Frey
Commandant (G-P-1/2)
USCG HQ
Washington, DC 20593\Dr. Sherrie Gott

Dr. Alinda Friedman
Department of Psychology
University of Alberta
Edmonton, Alberta
CANADA T6G 2E9

Dr. R. Edward Geiselman
Department of Psychology
University of California
Los Angeles, CA 90024

Dr. Michael Genesereth
Stanford University
Computer Science Department
Stanford, CA 94305

Dr. Dedre Gentner
University of Illinois
Department of Psychology
603 E. Daniel St.
Champaign, IL 61820

Dr. Jude Franklin
Code 7510
Navy Research Laboratory
Washington, DC 20375

Dr. Carl H. Frederiksen
McGill University
3700 McTavish Street
Montreal, Quebec H3A 1Y2
CANADA

Dr. John R. Frederiksen
Bolt Beranek & Newman
50 Moulton Street
Cambridge, MA 02138

Dr. Norman Frederiksen
Educational Testing Service
Princeton, NJ 08541

Dr. Daniel Gopher
Industrial Engineering
   and Management
TECHNION
Haifa 32000
ISRAEL

AFHRL/MODJ
Brooks AFB, TX 78235

Jordan Grafman, Ph.D.
Department of Clinical
   Investigation
Walter Reed Army Medical Center
6825 Georgia Ave., N. W.
Washington, DC 20307-5001

Dr. Wayne Gray
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. James G. Greeno
University of California
Berkeley, CA 94720

H. William Greenup (E031)
Education Center, MCDEC
Quantico, VA 22134

Dr. Robert Glaser
Learning Research
   & Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Marvin D. Glock
13 Stone Hall
Cornell University
Ithaca, NY 14853

Dr. Joseph Goguen
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Mr. William Hartung
PEAM Product Manager
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Reid Hastie
Northwestern University
Department of Psychology
Evanston, IL 60201

Dr. Harold Hawkins
Department of Psychology
University of Oregon
Eugene, OR 97403

Prof. John R. Hayes
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Barbara Hayes-Roth
Department of Computer Science
Stanford University
Stanford, CA 95305

Dr. Frederick Hayes-Roth
Teknowledge
525 University Ave.
Palo Alto, CA 94301

Ms. Glenda Greenwald, Editor
Human Intelligence Newsletter
P. O. Box 1163
Birmingham, MI 48012

Dipl. Pad. Michael W. Habon
Universitat Dusseldorf
Erziehungswissenschaftliches
Universitatsstr. 1
D-4000 Dusseldorf 1
WEST GERMANY

Dr. Henry M. Halff
Halff Resources, Inc.
4918 33rd Road, North
Arlington, VA 22207

Dr. Melissa Holland
Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Keith Holyoak
University of Michigan
Human Performance Center
330 Packard Road
Ann Arbor, MI   48109

Prof. Lutz F. Hornke
Universitat Dusseldorf
Erziehungswissenschaftliches
Universitatsstr. 1
Dusseldorf 1
WEST GERMANY

Mr. Dick Hoshaw
NAVOP-135
Arlington Annex
Room 2834
Washington, DC 20350

Dr. Steven Hunka
Department of Education
University of Alberta
Edmonton, Alberta
CANADA

Dr. Joan I. Heller
Graduate Group in Science and
 Mathematics Education
c/o School of Education
University of California
Berkeley, CA 94720

Dr. Jim Hollan
Code 51
Navy Personnel R & D Center
San Diego, CA 92152

Dr. John Holland
University of Michigan
2313 East Engineering
Ann Arbor, MI   48109

Dr. Daniel Kahneman
The University of British Columbia
Department of Psychology
#154-2053 Main Mall
Vancouver, British Columbia
CANADA V6T 1Y7

Dr. Milton S. Katz
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Steven W. Keele
Department of Psychology
University of Oregon
Eugene, OR 97403

Maj. John Keene
ADP Systems Branch
C3 Development Center (D104)
MCDEC
Quantico, VA 22134

Dr. Norman J. Kerr
Chief of Naval Education
 and Training
Code 00A2
Naval Air Station
Pensacola, FL 32508

Dr. Earl Hunt
Department of Psychology
University of Washington
Seattle, WA 98105

Dr. Ed Hutchins
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Zachary Jacobson
Bureau of Management Consulting
365 Laurier Avenue West
Ottawa, Ontario K1A 0S5
CANADA

Dr. Marcel Just
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. David Klahr
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Mazie Knerr
Program Manager
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Janet L. Kolodner
Georgia Institute of Technology
School of Information
 & Computer Science
Atlanta, GA 30332

Dr. Stephen Kosslyn
Harvard University
1236 William James Hall
33 Kirkland St.
Cambridge, MA 02138

Dr. Pat Langley
University of California
Department of Information
 and Computer Science
Irvine, CA 92717

Dr. David Kieras
University of Michigan
Technical Communication
College of Engineering
1223 E. Engineering Building
Ann Arbor. MI 48109

Dr. Peter Kincaid
Training Analysis
    & Evaluation Group
Department of the Navy
Orlando. FL 32813

Dr. Walter Kintsch
Department of Psychology
University of Colorado
Campus Box 345
Boulder. CO 80302

Dr. Jim Levin
University of California
Laboratory for Comparative
    Human Cognition
D003A
La Jolla. CA 92093

Dr. Michael Levine
Educational Psychology
210 Education Bldg.
University of Illinois
Champaign, IL 61801

Dr. Clayton Lewis
University of Colorado
Department of Computer Science
Campus Box 430
Boulder. CO   80309

Science and Technology Division
Library of Congress
Washington. DC 20540

Dr. Charlotte Linde
SRI International
333 Ravenswood Avenue
Menlo Park. CA 94025

Dr. Marcy Lansman
University of North Carolina
The L. L. Thurstone Lab.
Davie Hall 013A
Chapel Hill NC 27514

Dr. Jill Larkin
Carnegie-Mellon University
Department of Psychology
Pittsburgh PA 15213

Dr. Robert Lawler
Information Sciences FRL
GTE Laboratories inc
40 Sylvan Road
Waltham MA 02254

Dr. Alan M. Lesgold
Learning R&D Center
University of Pittsburgh
Pittsburgh PA 15260

Dr. Jay McClelland
Department of Psychology
Carnegie-Mellon University
Pittsburgh PA 15213

Dr. Kathleen McKeown
Columbia University
Department of Computer Science
New York. NY 10027

Dr. Joe McLachlan
Navy Personnel R&D Center
San Diego. CA 92152

Dr. James McMichael
Navy Personnel R&D Center
San Diego. CA 92152

Dr. Barbara Means
Human Resources
Research Organization
1100 South Washington
Alexandria. VA 22314

Dr. Arthur Melmed
U. S. Department of Education
724 Brown
Washington. DC 20208

Dr. Don Lyon
P. O. Box 44
Higley, AZ 85236

Dr. William L. Maloy (O2)
Chief of Naval Education
   and Training
Naval Air Station
Pensacola, FL 32508

Dr. Sandra P. Marshall
Department of Psychology
University of California
Santa Barbara, CA   93106

Dr. Manton M. Matthews
Department of Computer Science
University of South Carolina
Columbia, SC 29208

Dr. Richard E. Mayer
Department of Psychology
University of California
Santa Barbara, CA   93106

Dr. Allen Munro
Behavioral Technology
   Laboratories - USC
1845 S. Elena Ave., 4th Floor
Redondo Beach, CA 90277

Lt. Col. Jim Murphy
HQ, Marine Corps
Code MRRP
Washington, DC 20380

Director
Distribution Department
Naval Military Personnel Command
N-4
Washington, DC 20370

Head, HRM Operations Branch
Naval Military Personnel Command
N-62F
Washington, DC 20370

Dr. Al Meyrowitz
Office of Naval Research
Code 433
800 N. Quincy
Arlington, VA 22217-5000

Dr. Andrew R. Molnar
Scientific and Engineering
Personnel and Education
National Science Foundation
Washington, DC 20550

Dr William Montague
NPRDC Code 13
San Diego, CA 92152

Dr. Tom Moran
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

Headquarters, Marine Corps
Code MPI-20
Washington, DC 20380

Assistant for MPT Research,
Development and Studies
NAVOP 01B7
Washington, DC 20370

Head
Workforce Information Section
NAVOP 140F
Washington, DC 20370

Head
Manpower, Personnel, Training
   and Reserve Team
NAVOP 914D
5A578, The Pentagon
Washington, DC 20350

Leadership Management Education
   and Training Project Officer
Naval Medical Command (Code 05C)
Washington, DC 20372

Assistant for Evaluation.
   Analysis. and MIS
Naval Military Personnel Command
N-6C
Washington. DC 20370

Spec. Asst. for Research. Experi-
   mental & Academic Programs
Naval Technical Training Command
(Code 016)
NAS Memphis (75)
Millington, TN 38054

Program Manager for Manpower.
Personnel. and Training
NAVMAT 0722
Arlington. VA 22217-5000

Assistant for Long Range
   Requirements
CNO Executive Panel (NAVOP 00K)
2000 North Beauregard Street
Alexandria. VA 22311

Assistant for Planning MANTRAPERS
NAVOP 01B6
Washington. DC 20370

Director
Human Factors
   & Organizational Systems Lab
NPRDC (Code 07)
San Diego. CA 92152

Fleet Support Office
NPRDC (Code 301)
San Diego. CA 92152

Library
Code P201L
Navy Personnel R&D Center
San Diego. CA 92152

Mr. Bill Neale
HQ ATC/TTA
Randolph AFB. TX 78148

Technical Director
Navy Health Research Center
P.O. Box 85122
San Diego. CA 92138

Dr. Richard E. Nisbett
University of Michigan
Institute for Social Research
Room 5261
Ann Arbor. MI   48109

Dr. Donald A. Norman
Institute for Cognitive Science
University of California
La Jolla. CA 92093

Director. Training Laboratory
NPRDC (Code 05)
San Diego. CA 92152

Director. Manpower and Personnel
Laboratory
NPRDC (Code 06)
San Diego. CA 92152

Office of Naval Research
Code 433
800 N. Quincy Street
Arlington. VA 22217-5000

Office of Naval Research
Code 442EP
800 N. Quincy Street
Arlington. VA 22217-5000

Group Psychology Program
Code 442GP
Office of Naval Research
800 N. Quincy St.
Arlington. VA 22217-5000

Technical Director
Navy Personnel R&D Center
San Diego. CA 92152

Commanding Officer
Naval Research Laboratory
Code 2627
Washington. DC 20390

Dr. Harry F. O'Neil. Jr.
Training Research Lab
Army Research Institute
5001 Eisenhower Avenue
Alexandria. VA 22333

Dr. Stellan Ohlsson
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh. PA 15213

Director Technology Programs
Office of Naval Research
Code 200
800 North Quincy Street
Arlington. VA 22217-5000

Director Research Programs
Office of Naval Research
800 North Quincy Street
Arlington. VA 22217-5000

Mathematics Group
Office of Naval Research
Code 411MA
800 North Quincy Street
Arlington. VA 22217-5000

Dr. Douglas Pearse
DCIEM
Box 2000
Downsview. Ontario
CANADA

Dr. James W. Pellegrino
University of California,
    Santa Barbara
Department of Psychology
Santa Barbara. CA 93106

Office of Naval Research
Code 442PT
800 N. Quincy Street
Arlington. VA 22217-5000
(6 Copies)

Special Assistant for Marine
Corps Matters
Code 100M
Office of Naval Research
800 N. Quincy St.
Arlington. VA 22217-5000

Psychologist
ONR Branch Office
1030 East Green Street
Pasadena, CA 91101

Commanding Officer
Army Research Institute
ATTN: PERI-BR (Dr. J. Orasanu)
5001 Eisenhower Avenue
Alexandria. VA 22333

Prof. Seymour Papert
20C-109
Massachusetts Institute
    of Technology
Cambridge. MA 02139

Lt. Col. (Dr.) David Payne
AFHRL
Brooks AFB. TX 78235

Dr. Harry E. Pople
University of Pittsburgh
Decision Systems Laboratory
1360 Scaife Hall
Pittsburgh. PA 15261

Dr. Joseph Psotka
ATTN: PERI-1C
Army Research Institute
5001 Eisenhower Ave.
Alexandria. VA 22333

Dr. Nancy Pennington
University of Chicago
Graduate School of Business
1101 E. 58th St.
Chicago. IL 60637

Military Assistant for Training and
   Personnel Technology
OUSD (R & E)
Room 3D129, The Pentagon
Washington. DC 20301

LCDR Frank C. Petho, MSC. USN
CNATRA Code N36. Bldg. 1
NAS
Corpus Christi. TX 78419

Dr. Tjeerd Plomp
Twente University of Technology
Department of Education
P.O. Box 217
7500 AE ENSCHEDE
THE NETHERLANDS

Dr. Martha Polson
Department of Psychology
Campus Box 346
University of Colorado
Boulder. CO 80309

Dr. Peter Polson
University of Colorado
Department of Psychology
Boulder. CO 80309

Dr. Steven E. Poltrock
MCC
9430 Research Blvd.
Echelon Bldg #1
Austin. TX 78759-6509

Dr. Andrew M. Rose
American Institutes
   for Research
1055 Thomas Jefferson St.. NW
Washington. DC 20007

Dr. Lynne Reder
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh. PA 15213

Dr. James A. Reggia
University of Maryland
School of Medicine
Department of Neurology
22 South Greene Street
Baltimore. MD 21201

Dr. Fred Reif
Physics Department
University of California
Berkeley, CA 94720

Dr. Lauren Resnick
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh. PA 15213

Dr. Gil Ricard
Code N711
NAVTRAEQUIPCEN
Orlando. FL 32813

Dr. Mary S. Riley
Program in Cognitive Science
Center for Human Information
   Processing
University of California
La Jolla. CA 92093

William Rizzo
Code 712 NAVTRAEQUIPCEN
Orlando. FL 32813

Dr. Judah L. Schwartz
MIT
20C-120
Cambridge. MA 02139

Dr. Ernst Z. Rothkopf
AT&T Bell Laboratories
Room 2D-456
600 Mountain Avenue
Murray Hill, NJ 07974

Dr. William B. Rouse
Georgia Institute of Technology
School of Industrial & Systems
   Engineering
Atlanta, GA 30332

Dr. David Rumelhart
Center for Human
   Information Processing\
Univ. of California
La Jolla, CA 92093

Dr. Michael J. Samet
Perceptronics, Inc
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. Robert Sasmor
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Roger Schank
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Walter Schneider
University of Illinois
Psychology Department
603 E. Daniel
Champaign, IL 61820

Dr. Alan H. Schoenfeld
University of California
Department of Education
Berkeley, CA 94720

Dr. Marc Sebrechts
Department of Psychology
Wesleyan University
Middletown, CT 06475

Dr. Judith Segal
Room 819F
NIE
1200 19th Street N.W.
Washington, DC 20208

Dr. Ramsay W. Selden
U. S. Army Research Institute
5001 Eisenhower Avenue
Alexander, VA 22333

Dr. Ramsay W. Selden
NIE
Mail Stop 1241
1200 19th St., NW
Washington, DC 20208

Dr. Michael G. Shafto
ONR Code 442PT
800 N. Quincy Street
Arlington, VA 22217-5000

Dr. Sylvia A. S. Shafto
National Institute of Education
1200 19th Street
Mail Stop 1806
Washington, DC 20208

Mr. Colin Sheppard
Applied Psychology Unit
Admiralty Marine Technology Est.
Teddington, Middlesex
UNITED KINGDOM

Dr. Kazuo Shigemasu
7-9-24 Kugenuma-Kaigan
Fujusawa 251
JAPAN

Dr. Ted Shortliffe
Computer Science Department
Stanford University
Stanford, CA 94305

Dr. Lee Shulman
Stanford University
1040 Cathcart Way
Stanford, CA 94305

Dr. Robert S. Siegler
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Herbert A. Simon
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Zita M Simutis, Chief
Instructional Technology
    Systems Area
ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. H. Wallace Sinaiko
Manpower Research
    and Advisory Services
Smithsonian Institution
801 North Pitt Street
Alexandria, VA 22314

Dr. Edward E. Smith
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Alfred F. Smode
Senior Scientist
Code 7B
Naval Training Equipment Center
Orlando, FL 32813

Dr. Richard Snow
Liaison Scientist
Office of Naval Research
    Branch Office, London
Box 39
FPO New York, NY 09510

Dr. Elliot Soloway
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Richard Sorensen
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Kathryn T. Spoehr
Brown University
Department of Psychology
Providence, RI 02912

James J. Staszewski
Research Associate
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Frederick Steinheiser
CIA-ORD
612 Ames
Washington, DC 20505

Dr. Robert Sternberg
Department of Psychology
Yale University
Box 11A, Yale Station
New Haven, CT 06520

Dr. Albert Stevens
Bolt Beranek & Newman, Inc.
10 Moulton St.
Cambridge, MA 02238

Dr. Paul J. Sticha
Senior Staff Scientist
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Thomas Sticht
Navy Personnel R&D Center
San Diego, CA 92152

Dr. David Stone
KAJ Software. Inc.
3420 East Shea Blvd.
Suite 161
Phoenix. AZ 85028

Dr. John Tangney
AFOSR/NL
Bolling AFB. DC 20332

Dr. Martin M. Taylor
DCIEM
Box 2000
Downsview. Ontario
CANADA

Dr. Perry W. Thorndyke
FMC Corporation
Central Engineering Labs
1185 Coleman Avenue. Box 580
Santa Clara. CA 95052

Major Jack Thorpe
DARPA
1400 Wilson Blvd.
Arlington. VA 22209

Dr. Martin A. Tolcott
3001 Veazey Terr.. N.W.
Apt. 1617
Washington. DC 20008

Dr. Douglas Towne
Behavioral Technology Labs
1845 S. Elena Ave.
Redondo Beach. CA 90277

Dr. Amos Tversky
Stanford University
Dept. of Psychology
Stanford. CA 94305

Dr. James Tweeddale
Technical Director
Navy Personnel R&D Center
San Diego. CA 92152

Dr. Paul Twohig
Army Research Institute
5001 Eisenhower Avenue
Alexandria. VA 22333

Headquarters. U. S. Marine Corps
Code MPI-20
Washington. DC 20380

Dr. Kurt Van Lehn
Xerox PARC
3333 Coyote Hill Road
Palo Alto. CA 94304

Dr. Beth Warren
Bolt Beranek & Newman. Inc.
50 Moulton Street
Cambridge. MA 02138

Roger Weissinger-Baylon
Department of Administrative
   Sciences
Naval Postgraduate School
Monterey, CA 93940

Dr. Keith T. Wescourt
FMC Corporation
Central Engineering Labs
1185 Coleman Ave.. Box 580
Santa Clara. CA 95052

Dr. Douglas Wetzel
Code 12
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Barbara White
Bolt Beranek & Newman. Inc.
10 Moulton Street
Cambridge. MA 02238

Dr. Mike Williams
IntelliGenetics
124 University Avenue
Palo Alto. CA 94301

Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Martin F. Wiskoff
Navy Personnel R & D Center
San Diego. CA 92152

Dr. Frank Withrow
U. S. Office of Education
400 Maryland Ave. SW
Washington. DC 20202

Dr. Merlin C. Wittrock
Graduate School of Education
UCLA
Los Angeles. CA 90024

Mr. John H. Wolfe
Navy Personnel R&D Center
San Diego. CA 92152

Dr. Wallace Wulfeck. III
Navy Personnel R&D Center
San Diego. CA 92152

Dr. Joe Yasatuke
AFHRL/LRT
Lowry AFB. CO 80230

Major Frank Yohannan. USMC
Headquarters. Marine Corps
(Code MPI-20)
Washington. DC 20380

Mr. Carl York
System Development Foundation
181 Lytton Avenue
Suite 210
Palo Alto. CA 94301

Dr. Joseph L. Young
Memory & Cognitive
    Processes
National Science Foundation
Washington. DC 20550

Cmdr. Joe Young
HQ. MEPCOM
ATTN: MEPCT-P
2500 Green Bay Road
North Chicago. IL 60064

Dr. Steven Zornetzer
Office of Naval Research
Code 440
800 N. Quincy St.
Arlington, VA 22217-5000

END

FILMED

10-85

DTIC